A Droplet Routing Technique for Fault-Tolerant Digital Microfluidic Devices

Xiao Zhang, Frits van Proosdij, and Hans G. Kerkhoff
Testable Design and Test of Integrated Systems Group, CTIT
University of Twente, Enschede, the Netherlands

Abstract—Efficient droplet routing is one of the key approaches for realizing fault-tolerant microfluidic biochips. It requires that run-time diagnosis and fault recovery can be made possible in such systems. This paper describes a droplet routing technique for a fault-tolerant digital microfluidic platform. This technique features handling of many microfluidic operations simultaneously and uses on-chip sensors for diagnosis at run-time. Once a fault is detected during the droplet routing, recovery procedures will be started-up immediately. Faulty units on the chip will be marked and isolated from the array so that the remaining droplets can still be routed along a fault-free path to their destinations. This method guarantees a non-stop fault-tolerant operation for very large microfluidic arrays.

I. INTRODUCTION

The rapid advance in microfluidic devices enables the possibility to perform conventional biological laboratory activities on a *portable* platform. The miniature size of these devices leads to shorter detection times, high throughput and reduction on energy and reagents consumption in biological experiments. A new generation of microfluidic devices known as the "digital microfluidic biochip" was introduced to the community by using the electrowetting-on-dielectric principle [1]. Small nano-litre volume droplets can be manipulated on a planar surface of the microfluidic biochip and further more complex operations such as splitting and merging droplets were realized too [2].

Droplet routing on the surface of the microfluidic biochip has been attracting a lot of attention in recent years as it is one of the key issues to make use of the digital microfluidic device efficiently. Topics such as optimal droplet routing [3] and scheduling of microfluidic operations [4] have been studied. Furthermore, simulations have been carried out to monitor the behaviour of multiple assays running simultaneously on the same biochip platform [5]. It has been investigated and proven that during droplet routing, specific faults may occur due to surface contamination, dielectric layer break down or shorting between control electrodes [6, 7]. Some claim that digital microfluidic biochips should be made simple and cheap so that they can be disposed of once faults appear. However, for certain applications like peptide production platforms, it is necessary to combine very large scale microfluidic arrays with complex control, sensing and signal processing electronics on a single chip by using advanced CMOS technology [6]. In these cases it is not recommended to discard the complete biochip due to only some defects. In these applications, diagnosis and "repairing" the digital microfluidic biochips become an essential issue.

In previous research, functional testing of the biochips and fault models have been proposed [7]. Parallel scan-like tests have also been used to enhance the diagnosis efficiency [8]. However, run-time diagnosis and fault recovery have not been thoroughly studied before and they are of key importance for fault-tolerant digital microfluidic devices.

In this paper a droplet routing technique is proposed to address the problems mentioned above. In Section II, the background knowledge of digital microfluidics is briefly introduced and our experimental platform is explained. In Section III, an application example, being peptide synthesis, and the related droplet routing problems are described. Then in Section IV, the routing method together with fault diagnosis and recovery are discussed in detail. Section V provides the software implementation flow chart and the simulation results. Finally, concluding remarks are provided in Section VI.

II. THE DIGITAL MICROFLUIDIC DEVICE PLATFORM

The basic building block of our digital microfluidic device is shown in 0. It employs the principle of electrowetting on dielectric [1]. The bottom plate contains a patterned array of Chromium control electrodes; each of them can be separately controlled and conductive ITO glass is used as the top plate. The droplet is in the middle of the two plates separated by tiny spacers to form the channel; the channel is coated with a hydrophobic layer. If the electrodes underneath are correctly activated, the droplet in the middle will move along the Cr electrodes. The structure includes multiple direct Platinum (Pt) droplet sensing electrodes as well as indirect capacitive Cr sensing electrodes.
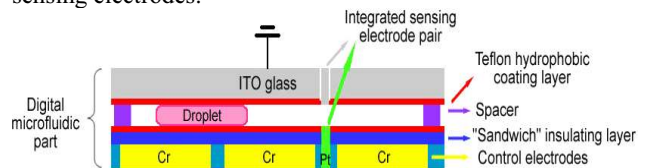

Figure 1: Basic cell used in an EWOD-based digital microfluidic biochip

A prototype digital microfluidic platform was designed and implemented and is shown in 0. A NIOS processor and all control software were incorporated onto the FPGA

development board (top Figure 2). The routing of droplets is being computed on the FPGA by the NOIS processor and resulting low-voltage control signals were generated by the FPGA output pins. These were subsequently amplified by a DMOS level-shifter circuit (bottom left). The high voltage signals were applied to the PCB with mounted microfluidic chip to control the droplets. A detail of the microfluidic chip is depicted at the right of Figure 2. More details can be found in [9].
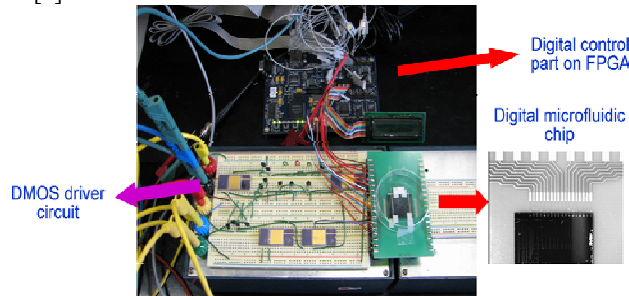

Figure 2: Our initial prototype digital microfluidic platform [10]

III. PEPTIDE SYNTHESIS ON A DIGITAL MICROFLUIDIC PLATFORM AND THE ROUTING CHALLENGE

A. Peptide Synthesis on our Digital Microfluidic Device

A peptide is a chain of many types of amino acids of a certain length. It plays a crucial role in body health of mankind and in modern medicine research or medical point-of-care diagnosis. The basic principle for forming a peptide is that two amino acids are linked by a peptide bond (also referred as "amide bond"). This process is usually called "peptide coupling". New amino acids are added to the formed chain until the target peptide is complete. In industry, dedicated peptide synthesis machines are being used to synthesize new peptides but these machines are very expensive to purchase and not very flexible. In practice, scientists in labs often need to synthesize many types of peptide in just a small quantity to compare their properties. It is preferable if the synthesis of different peptides can take place in parallel on the same platform and in a small quantity.

In previous papers [9]-[11] we have suggested and proven the required interconnection implementation of a new droplet-based peptide synthesizer using a new advanced heterogeneous technology, combining automotive SoC and microfluidics technology. The "on-chip" peptide synthesizer works in a similar way as the traditional industrial machines. First, the target peptide sequence can be inputted by the user. Then droplets are used to carry the amino acids and add them to the peptide chain under synthesis. The digital microfluidic device may consist of hundreds of basic cells (one cell represents a controllable part on the device for moving the droplet) and they are divided into several functional zones for reagent storage, droplet transportation and reaction, etc. 0 shows an example of a 16x16 cell array demonstrating

different functional zones and possible droplet manipulation paths.

The cell array can be abstracted as a 2D diagram and the cells can be numbered according to their coordinates. The electrode in the upper left corner is labelled (0, 0), the right lower corner is referred to as (15, 15). Movable droplets are represented as brick-shaded balls. The current position of a droplet and its source and destination for its transportation can all be accessed by the central control software which takes care of the manipulation of all the droplets in the array at the same time. The shortest path for each travelling droplet is being computed by our developed algorithm. The solid dark gray zones represent faulty cells and are not allowed to be used. Dashed black and point gray lines both show the paths a droplet can choose to arrive at its destination. The black-boxed cells indicate that a droplet presence sensor is available in that cell.
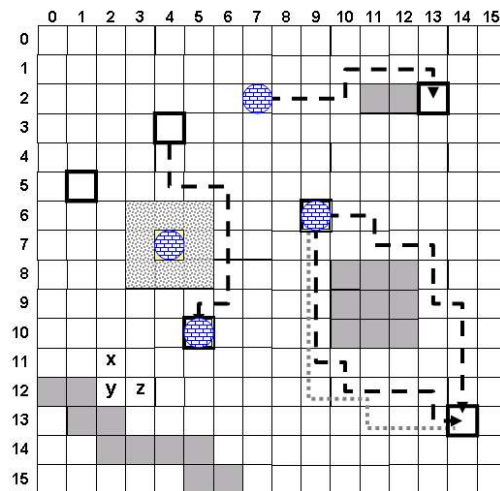

Figure 3: A 16x16 cell array example [12]

B. The Routing Challenge

Because droplets are more than abstract balls, certain constraints have to be applied while moving the droplets on the device. For proper operation, the size of a droplet is slightly larger than the size of one (electrode) cell on the microfluidic device, so no droplets can be present in the nearby cells of an existing droplet. For example, the 8 nearby cells around the droplet at (4, 7) are marked speckled light gray, indicating no droplets should enter the "speckled light gray"; otherwise an unwanted droplet merging might take place.

During normal operation, certain cell(s) on the array may become faulty, preventing the droplets from arriving on their expected destination as verified in other papers [8]. The possible defects in a digital microfluidic device and their fault models have been well summarized and proven in [7]; in terms of fault coverage in our fault-tolerance case it means that all targeted (single) defects are being covered.

Most of the faults models fall into the following three categories which have been physically proven to exist in practice:

- *Dead cells*. The electrode can not be activated anymore, or some other reason causes this cell to remain empty.
- *Sticky cells*. The electrode is stuck-on, the hydrophobic layer is contaminated or some other reason causes the droplets to stay in this cell. The sticky cell also causes the following problem: as soon a new droplet enters the 8 neighbour cells of the sticky cell with a droplet already present, the new droplet will be merged with the present droplet. If left unattended by the control system, the sticky-cell zone will grow even larger.
- *Two shorted electrodes*. A conducting particle may connect two nearby electrodes. This will cause the droplet being left in the middle of the two electrodes and will be stuck in the middle. It behaves similarly to sticky cells thus will not be discussed separately.

The challenge of droplet routing in the digital microfluidic device is when faults appear; how to detect them at run time and effectively isolate the faulty area so that the least usable cell area is wasted. At the same time, the normal operation of the device should still be maintained (fault-tolerant behaviour). It is not required to hold or cancel the many on-going operations just because a fault occurred.

Our solution to the run-time detection of the fault is to keep track of the droplets in the array while moving them. This is achieved by placing the so called "check cells" on the critical positions of the droplet moving path, such as at the beginning and the ending cell. The electrode used to move the droplet can also be reused as a sensor for droplet presence (and content) sensing [10] [11]; hence they are used as "check cells". If a target droplet does not arrive at the correct check cell at the expected time, the conclusion that a fault has occurred can be drawn.

IV. ROUTING FOR A FAULT-TOLERANT DIGITAL MICROFLUIDIC DEVICE

A. The Basic Routing Idea

The control software requires three inputs to guarantee successful (and hence fault-tolerant) droplet transportation. First, it needs to know which cells on the array are available for usage. Second, it requires data with regard to the droplet's starting and ending cell. Third, it needs an acknowledge message from the check cell to ensure the transport was successfully achieved.
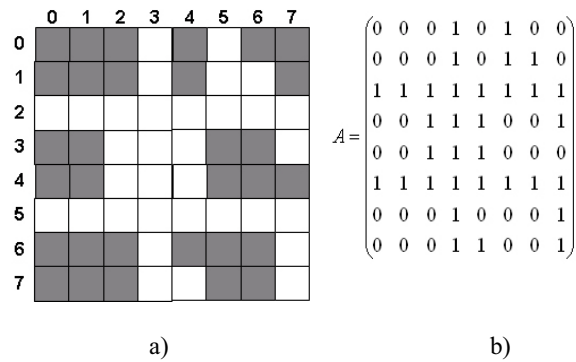

Figure 4: The cell array and its equivalent matrix [13]

Figure 4a shows an example of the cell array with white cells indicating usable ones and dark grey ones as faulty or occupied ones (have droplet on it or nearby). A matrix is created by the control software to record the status of each cell on the array, in which "0" represents unusable cells and "1" the usable ones (Figure 4b). This matrix is updated whenever the location of the droplets on the array change.

Given the starting and ending point and the available cells on the array, the "shortest path finding" algorithm was developed which can compute the shortest route for the droplet; subsequently, the droplet will be moved along its path. Since it is known in advance how many steps it takes for the droplet to arrive at its destination, and hence check cell, one can easily verify if the droplet has arrived at the proper time.

B. Run-Time Fault Diagnosis

If a droplet does not arrive at the check cell as expected, some fault could have occurred along the required path and the droplet is left somewhere along the path. For the ease of study, two assumptions are made:

- only <u>one</u> fault will occur for one droplet during its transportation
- the fault will only occur along the path the droplet travels

One example is shown in Figure 5. The speckled droplet is planned to be moved from cell a to cell j along the black dashed path. But cell g is assumed to be a *dead cell* (indicated by ellipse cross). In this case, the droplet will be stuck at cell f since cell g can not be activated and thus can not pull the droplet from cell f to cell g. The droplet will not arrive at its destination cell j at the expected time. But the droplet still can move from cell f to cell e.

Another possibility is that cell f is a *sticky cell*; in this case the droplet will be stuck at cell f and can not be moved to any direction anymore.
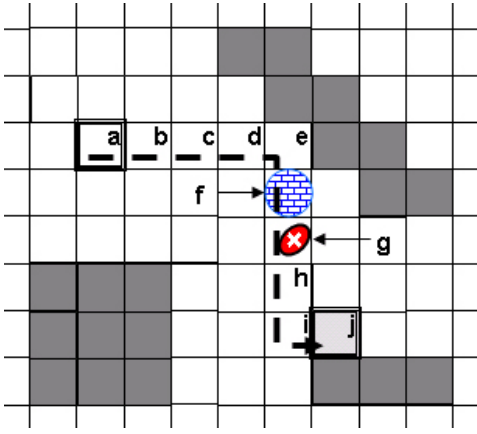
Figure 5: Example of droplet on cell f in case cell g is a dead cell [12]

The control software, however, has no knowledge of the situation as shown in Figure 5; yet it still needs to determine where the droplet was lost and subsequently try to mask the faulty area to prevent further use.

The key issue for solving this problem is to look for the missing droplet. If the faulty cell g is a dead cell, then the software can try to route the droplet back to cell a. This can be accomplished as follows: the first suspicious cell is cell b and if cell b is a dead cell, the droplet would be left on cell a; this causes the first sensor at the beginning of the route to report correct droplet movement and hence concluding cell b has a fault-free operation. Now, cell c may be defective. This would mean that the droplet is still on cell b. By activating cell a to pull from cell b, one can verify if this is the case. Following this method the next suspect is cell d. Cells b and cell a are activated sequentially, causing any droplet in cell c to move from c to a. Cell a is examined on droplet presence, but no droplet can be sensed in cell a in our case. This indicates that cell d operates fault-free. This method can be used for all suspicious cells until the dead cell g is found. The time it takes to find a dead cell varies according to the location of the dead cell. Cell b is checked in 1 time step. Cell c is checked in 2 time steps, the 3rd cell is checked in 3 time-steps, finally the (L-1)th cell is checked in (L-1) time steps. Cell L (the last cell) is checked in L steps. The time it takes to test the whole route is:

$$t_{max} = \sum_{s=1}^{L} s = (L+1)\left(L/2\right) = \frac{1}{2}L^2 + \frac{1}{2}L \quad (1)$$

To reduce this time, it is possible to use pipelining. Investigation of cell e (the 4th cell) can start before the checking of cell d is finished, because when the content of cell d is transported from b to a, in the meantime, the content of e can be transported to d. This is why it will take 3 steps for each of the 4th up to the Lth cell to check their content. The parameter x in t_x indicates that there are x cells between the

starting cell and the dead cell. This is why the total time it takes to check for dead cells is:

$$\begin{aligned} t_1 &= 1 \\ t_2 &= 2 \\ t_3 &= 3 \\ t_{n>3} &= 3 \\ t_L &= 3L - 3 \end{aligned} \quad (2)$$

After checking cell f, which is fault-free, cell g is ready to be investigated. Since cell g is a dead cell, the droplet will remain at its location in cell f. Cells a up to e are working correctly, so the droplet from cell f can be transported to electrode a. The droplet will now be routed back to cell a where it can be detected and it can be concluded that cell g is a dead cell.

If the faulty cell f is a sticky cell, the droplet will be stuck on it and will not move again regardless which cell is activated along the green path. In this case, the method used for the dead cell will not work anymore. Hence, a new method which sends out a *test droplet* will be used.

First, a test droplet is routed to cell a (refer to Figure 50) from a test-droplet source (not shown here for simplicity). If cell b is the sticky cell, the test droplet will be merged with the droplet stuck on b and it can not get back to cell a again. If cell b is not a sticky cell, one starts with testing cell c. The test droplet is now transported to cell b, and back to cell a. If it arrives at cell a, at least cell c is not sticky. Then, cell d is tested, where the droplet is transported (via cell b) to cell c, and back (via cell b) to cell a. If the previous is verified to be correct, cell d operates fault-free; otherwise it would have kept away the test droplet from cell c. Following this method, all cells can be checked. The time it takes to find the sticky cell is:

$$\begin{aligned} t_1 &= 2 \\ t_n &= 2n \\ t_L &= \sum_{s=1}^{L-1} 2s \\ t_L &= L^2 - L \end{aligned} \quad (3)$$

In practice, it may be more efficient to send another test droplet from the other end of the route at the same time if possible. The time it takes to find the sticky cell may be shortened significantly. In this case equation (4) holds.

C. Fault Recovery

Once faulty cells are located, it is important to mask them and prevent further use of these cells.

In the case of a *dead cell*, the cell has to be masked as unavailable but the cells around it can still be put into further use. In Figure 3, for example, dead cell g will be assigned a "0" to its corresponding position in the array matrix (0).

.

$$t = 2\sum_{s=1}^{\left\lceil (L-1)/2 \right\rceil} s$$

$$t_{oddL} = 2\sum_{s=1}^{(L-1)/2} s \qquad (4)$$

$$t_{oddL} = \frac{1}{4}L^2 - \frac{1}{4}$$

$$t_{evenL} = 2\sum_{s=1}^{L/2} s$$

$$t_{evenL} = \frac{1}{4}L^2 + \frac{1}{2}L$$

In the case of a *sticky cell*, the droplet will always be staying at the sticky location and there is simply no way to remove it. Moreover, the 8 nearby cells of that sticky cell have to be masked as unavailable too. Therefore nine "0"s will be added to the array matrix in the corresponding locations.

In the case of *shorted cells*, the 2 shorted cells should be disabled and the 10 surrounding cells around them should be disabled. And 10 "0"s will be added to the array matrix.

By continuously updating the array matrix, the central control software always knows which location on the array is unavailable for droplet transportation thus avoiding the droplets to run into faulty areas and also prevents them from running into each other

V. IMPLEMENTATION

The droplet routing control flow chart is shown in Figure 6. The software performs the following basic steps:

- Transport the droplet following the given route
- With regard to each check cell: verify whether the droplet has arrived or not.
- Check for dead cells
- Check for sticky and shorted cells
- Mark faulty cells.

A. Verify the Existence of Dead Cells

There is no test droplet necessary to look for a dead cell. One may assume that the first and the last cell on the route are check cells. An example cell-activation sequence diagram is presented in Figure 7. The left column indicates the time step sequence; the top line shows the cell number along the droplet transportation path and the arrows indicates the movement of a possible droplet.

B. Verify the existence of Sticky Cells

Once it is investigated that there are no dead cells, there could be a sticky cell along the route. 0 shows the cell activation sequence to look for the sticky cell from both sides of the route.
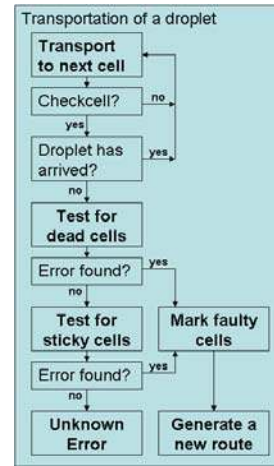

Figure 6: Flow chart of the droplet routing scenario of control software [12]

Time step	0	1	2	3	4	…	L-1	L
0		←						
0	?							
1			←					
2		←						
3	?			←				
4			←					
5		←			←			
6	?			←				
7			←					
8		←						
9	?							

Figure 7: Cell activation sequence to verify the existence of a dead cell

Time step	0	1	2	3	4	5	…	L-1	L
	?								?
1	→							←	
1	←							→	
	?								?
2	→							←	
2		→					←		
2		←					→		
2		←						→	
	?								?
3	→							←	
3		→					←		
3			→		←				
3			←		→				
3		←					→		
3	←							→	
	?								?

Figure 8: Cell activation sequence to verify the existence of sticky cell

To examine the routing program and the central control software, a test bench and a graphical user interface has been developed in the C programming language. The user can manually create a cell array layout with some defects on it and then introduce a target peptide to be synthesized. It can be observed from the simulation results that correct amino acids for synthesis were chosen and successfully routed to the reaction zone.

A simplified routing problem is shown here to verify the routing software. In Figure 9a, the layout of a digital microfluidic array is shown. The numbers on the top and left side indicate the x and y coordinates of the cells. Droplets on the array are represented by 45 degrees shaded squares. White cells are fault-free ones and can be used for droplet transportation and the horizontal shaded square cells are known faulty ones. The symbol "[ ]" indicates that a droplet presence sensor is available in that cell. The symbol "##" denotes a droplet reservoir.

Suppose it is required to route a droplet from location (6, 1) to (3, 6). One can introduce the starting and ending coordinates and an optimal route is calculated by the program, see 0(a). Then on this route, we assume a dead cell at (6, 6) to test the program's fault diagnosis capability.
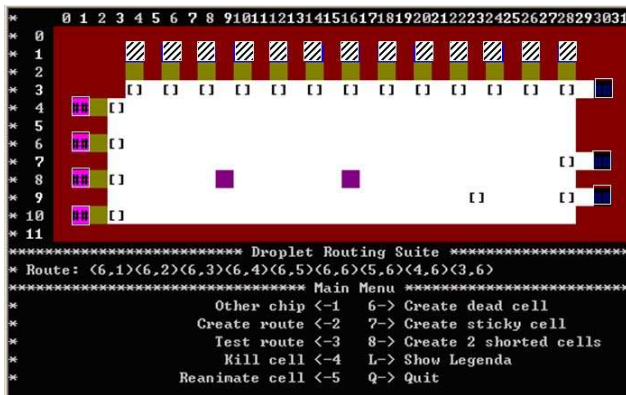

Figure 9a: Array layout and calculation results of the optimal route

When the simulation process was started, the calculated route was depicted as light grey in the array in Figure 9b. The red square (6, 7) represents the cell whose electrode was activated. It can be seen in the simulation window that the droplet (45 degrees hatched square) was stuck at (6, 5) and the electrode activation has lost its control on the droplet because (6, 6) is a dead cell.

At the end of the routing process, the droplet presence sensor at (3, 6) will indicate that no droplet has arrived and thus a fault could have occurred along the route. Then the diagnosis process starts the fault-detection procedure. Using the method proposed above, the program succeeded in finding the dead cell location as seen in Figure 9c.
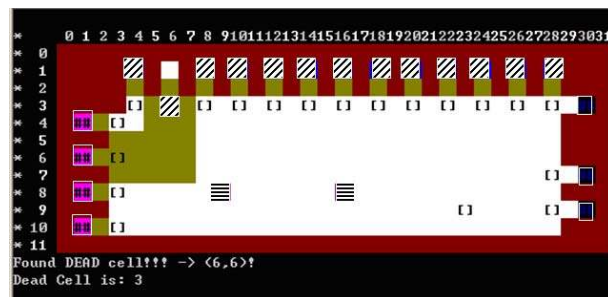

Figure 9b: Droplet routing process simulation

Figure 9c: Fault diagnosis found the dead cell location

Figure 9: Complete software simulation of the routing process

VI. CONCLUSIONS

This paper presented a novel method for routing droplets on a digital microfluidic device to ensure a fault-tolerant behavior. Fault models on this type of device have been examined and three typical faults were targeted. Problems such as optimal routing, and now also run-time fault diagnosis and subsequent fault recovery, have been discussed and efficient solutions have been given. Test time has not been our major concern as it is performed on-board.

Proper locations for checkcells have been suggested so that only a small number of droplet presence sensors are necessary to be incorporated in the device. A tool-set and experiment platform to evaluate peptide routing under fault-free and faulty cells conditions have been developed. The routing and diagnostic software has also been developed in the C programming language and simulations have been carried out to verify the routing and fault diagnosis capability of the software. The results were encouraging. The programs have been implemented on the designated local processor on the FPGA board.

REFERENCES

[1] M.G. Pollack, R.B. Fair, and A.D. Shenderov, "Electrowetting-based actuation of liquid droplets for microfluidic applications," Applied Physics Letters, Vol. 77, No.11, pp. 1725-1726, 2000.

[2] S. K. Cho, H. Moon, and C-J. Kim, "Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits," Journal of Microelectro-mechanical Systems, vol.12, no.1, pp. 70-80, Feb 2003.

[3] K.F. Bohringer, "Towards optimal strategies for moving droplets in digital microfluidic systems," IEEE International Conference on Robotics and Automation, vol. 2, pp. 1468-1474, April 2004.

[4] J. Ding, K. Chakrabarty, and R.B. Fair, "Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.20, no.12, pp. 1463-1468, Dec 2001.

[5] E.J. Griffith, S. Akella, and M.K. Goldberg, "Performance Characterization of a Reconfigurable Planar-Array Digital Microfluidic System," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 2, pp. 345-357, Feb. 2006.

[6] H.G. Kerkhoff, "Testing Micro-electronic Biofluidic Systems", IEEE Design and Test of Computers, pp. 72-82, January 2007.

[7] T. Xu, and K. Chakrabarty, "Functional testing of digital microfluidic biochips," IEEE International Test Conference, pp.1-10, Oct 2007.

[8] 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, pp. 148-158, June 2007.

[9] H.G. Kerkhoff and X. Zhang, "Fault Simulation of Heterogeneous Integrated Biological Systems", in proc. IEEE IMSTW, Edinburgh, Scotland, pp. 121-129, 2006.

[10] BioDrop Progress report, Editor H.G. Kerkhoff, DfMM NoE PATENT, August 167 pages, 2007.

[11] H.G. Kerkhoff et al., "A Dependable Microelectronic-Fluidic Peptide Synthesizer using Electrode Data", VLSI Design, Hindawi Publishers, 2008.

[12] J.F. van Proosdij, "Software for detection of faulty cells in microfluidic chips", Individual Project Report, #6920072, University of Twente, Dec 2007.

[13] T. Ran et al., "Automatic Drug Synthesis", B2 Project Report, #6920071, University of Twente, Jul 2007.