Defect-Tolerant Design and Optimization of a Digital Microfluidic Biochip for Protein Crystallization

Tao Xu, *Student Member, IEEE,* Krishnendu Chakrabarty, *Fellow, IEEE,* and Vamsee K. Pamula

***Abstract*—Protein crystallization is a commonly used technique for protein analysis and subsequent drug design. It predicts the 3-D arrangement of the constituent amino acids, which in turn indicates the specific biological function of a protein. Protein crystallization experiments are typically carried out in well-plates in the laboratory. As a result, these experiments are slow, expensive, and error-prone due to the need for repeated human intervention. Recently, droplet-based "digital" microfluidics have been used for executing protein assays on a chip. Protein samples in the form of nanoliter-volume droplets are manipulated using the principle of electrowetting-on-dielectric. We present the design of a multi-well-plate microfluidic biochip for protein crystallization; this biochip can transfer protein samples, prepare candidate solutions, and carry out crystallization automatically. To reduce the manufacturing cost of such devices, we present an efficient algorithm to generate a pin-assignment plan for the proposed design. The resulting biochip enables control of a large number of on-chip electrodes using only a small number of pins. Based on the pin-constrained chip design, we present an efficient shuttle-passenger-like droplet manipulation method and test procedure to achieve high-throughput and defect-tolerant well loading.**

***Index Terms*—Digital microfluidics, droplet routing, lab-on-chip, pin-constrained biochip design, route scheduling, well-plate chip.**

I. Introduction

PROTEINS play a key role in all biological processes. The specific biological function of a protein is determined by the 3-D arrangement of the constituent amino acids. Therefore, the 3-D structure of a protein needs to be understood for effective protein engineering, bioseparations, rational drug design, controlled drug delivery, as well as the design of novel enzyme substrates, activators, and inhibitors [1], [2]. A widely used method to study the 3-D structure of proteins is to crystallize the proteins and determine the structure using X-ray diffraction [3].

Studies have been reported in the literature to gain a fundamental understanding of the mechanism of crystallization [4]. However, due to the complexity of the process and the number of parameters involved, it may take years before the process is understood well enough to have practical value. However, structural biologists need immediate information about the structures of proteins; hence, empirical methods are widely employed for crystallization. For example, an empirical approach that is typically used, among others, is a 2-D coarse sampling that involves systematic variation of salt concentration versus pH [4].

Proteins are crystallized in mainly four different ways: batch, vapor diffusion, liquid/liquid (free interface) diffusion, and dialysis methods [5]. We focus here on batch crystallization methods, where the protein to be crystallized (with "bench-top" volumes ranging from 50 μl to 1 ml) is mixed with the crystallizing agents at the required concentration at the start of the experiment. In this case, supersaturation is reached immediately upon mixing. Protein crystallization is a multi-parametric process that involves the steps of nucleation and growth, where molecules are brought into a thermodynamically unstable and a supersaturated state. In order to "hit" upon the correct parameters for the crystallization of proteins, a large number of experiments (10^3–10^4) are typically required, which leads to the consumption of substantial protein volumes and long time durations.

Efforts are ongoing to reduce the consumption of proteins by miniaturizing the crystallization setup. Screening for protein crystallization includes many repetitive and reproducible pipetting operations. To ease this manual and time-consuming task, several automatic methods have been introduced. In 1990, Chayen *et al.* introduced a micro-batch method where only 1 μl of protein and 1 μl of precipitants were dispensed by programmed Hamilton syringes [6] in each well of a 96-well-plate containing paraffin oil. Microbatch crystallization has been recently demonstrated in micropipettes in 1 μl droplets by DeTitta's group at Hauptman Woodward Institute (HWI), where the precipitant and the protein solutions are loaded manually into a microcentrifuge tube, centrifuged, collected in a micropipette, and then sealed [7]. Despite the efforts at reducing protein volumes, these processes still consume

Manuscript received June 5, 2009; revised October 12, 2009. Current version published March 19, 2010. This work was supported in part by the National Institute of General Medical Sciences of the National Institute of Health, under Grant No. R44GM072155, and the National Science Foundation, under Grant Nos. CCF-0541055 and CCF-0914895. A preliminary version of this paper was published in the Proceedings of the International Conference on Computer-Aided Design, 2008. This paper was recommended by Associate Editor, N. K. Jha.

T. Xu is with Cisco Systems, Inc., Research Triangle Park, NC 27709 USA (e-mail: txu2@cisco.com).

K. Chakrabarty is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: krish@ee.duke.edu).

V. K. Pamula is with Advanced Liquid Logic, Inc., Research Triangle Park, NC 27560 USA (e-mail: vpamula@liquid-logic.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2042888

a significant amount of protein samples and they are labor-intensive.

Robotic automation has emerged as the dominant paradigm in the state-of-the-art high-throughput protein crystallization. However, robots are expensive and they require high maintenance. Currently, there exist only a few automatic crystallization systems that are commercially available. Douglas Instruments' Oryx 8 [8] can perform both microbatch and vapor diffusion methods on protein samples in the range of 0.1–2 μl. Gilson's robotic workstations [9] can also perform both microbatch and vapor diffusion on protein samples of about 1 μl. Syrrx, a rational drug design company, manufactures a robotic system [10] for protein crystallization utilizing 20 nL to 1 μl protein samples. State-of-the-art robotic systems at HWIs NIH-funded Center for High-Throughput Crystallization have a throughput of 69 000 experiments per day for setting up microbatch crystallization conditions, i.e., a 96 well-plate can be setup every 2 min. However, each screening condition still requires 0.4 μl of protein. These semi-automatic systems do not encompass ideal high-throughput configurations, requiring user intervention for multiple tray processing, as well as suffering from other material processing issues. As most of the work performed with these systems is not on a large scale, the automation of storage and handling of plates was not addressed in these systems [11]. Such industrial systems, even though they are capable of setting up thousands of crystallization screens a day, are prohibitively expensive for academic research labs, e.g., Douglas Instruments' Oryx 8 costs $65 300 per platform [8]. Therefore, affordable and high-throughput automation methods are still needed.

Recent studies have focused on the application of a high-throughput and inexpensive technology, referred to as digital microfluidics, to protein assays. Digital microfluidics is an emerging technology that aims to integrate fluid-handling on a chip. Bioassay protocols are scaled down (in terms of liquid volumes and assay times), and run on a microfluidic chip by manipulating discrete droplets of nanoliter volumes using a patterned array of electrodes [13]. By reducing the rate of sample and reagent consumption, digital microfluidic biochips enable continuous sampling and analysis for on-line, real-time, chemical and biological analysis, which make it uniquely suitable for high-throughput protein crystallization [14]. Computer-aided design tools for digital microfluidic biochips have emerged recently [15]–[22].

Recent studies have also shown the feasibility of carrying out protein crystallization on a digital microfluidic biochip. In [23], Srinivasan *et al.* presented a fabricated digital microfluidic biochip for protein stamping, which is capable of handling transportation and mixing of droplets enclosing protein samples with concentrations up to 0.01 mg/ml. The implementation of the basic operations in protein crystallization clearly highlights the promise of a protein crystallization biochip that relies on digital microfluidics. However, no automated chip design technique has thus far been proposed.

In this paper, we present a prototype design of a multi-well-plate biochip for protein crystallization. The chip layout consists of 96 wells for high-throughput processing. To reduce control complexity and fabrication cost, an efficient pin-assignment and control scheme is also proposed and applied to the design. In this way, a large number of on-chip electrodes can be controlled using a small number of control pins, with minimal impact on the system throughput. Based on the pin-constrained chip design, we present an efficient shuttle-passenger-like droplet manipulation method to achieve high-throughput and defect-tolerant well loading. A testing and diagnosis technique for locating defects is also presented.

The rest of the paper is organized as follows. Section II provides an overview of digital microfluidic biochips. In Section III, we introduce the 96-well-plate chip design. In Section IV, an efficient "Connect-5" [24] algorithm is used to generate pin-assignment plan for the proposed design, which enables efficient control of the biochip with only a small number of pins. We show how the Connect-5 can be applied on the proposed chip with irregular electrode layout. We also show how the chip can be designed for the manipulation of 2x volume droplets and how diagonal electrode interference can be avoided. In Section V, we describe the droplet-routing algorithm for efficient well-loading. In Section VI, we discuss the application of a modified parallel scan-like test method to the proposed chip design. Section VII shows how defect tolerance can be achieved. The droplet-routing and defect-tolerance methods are evaluated in Section VIII. Finally, conclusions are drawn in Section IX.

II. Digital Microfluidics

A digital microfluidic biochip utilizes the electrowetting phenomenon to manipulate and move nanoliter droplets containing biological and chemical samples on a 2-D electrode array [25]. A unit cell in the array includes a pair of electrodes that acts as two parallel plates. The bottom plate contains a patterned array of individually controlled electrodes, and the top plate is coated with a continuous ground electrode. A droplet rests on a hydrophobic surface over an electrode, as shown in Fig. 1. It is moved by applying a control voltage to an electrode adjacent to the droplet and, at the same time, deactivating the electrode just under the droplet. This electronic method of wettability control creates interfacial tension gradients that move the droplets to the charged electrode. Using the electrowetting phenomenon, droplets can be moved to any location on a 2-D array.

By varying the patterns of control voltage activation, many fluid-handling operations such as droplet merging, splitting, mixing, and dispensing can be executed in a similar manner. For example, mixing can be performed by routing two droplets to the same location and then turning them about some pivot points. The digital microfluidic platform offers the additional advantage of flexibility, referred to as reconfigurability, since fluidic operations can be performed anywhere on the array. Droplet routes and operation schedules are programmed into a microcontroller that drives electrodes in the array. In addition to electrodes, optical detectors such as light-emitting diodes and photodiodes are also integrated in digital microfluidic arrays to monitor colorimetric bioassays [23]. A film (filler fluid) of silicone oil is typically used to prevent contamination and cross contamination [25].

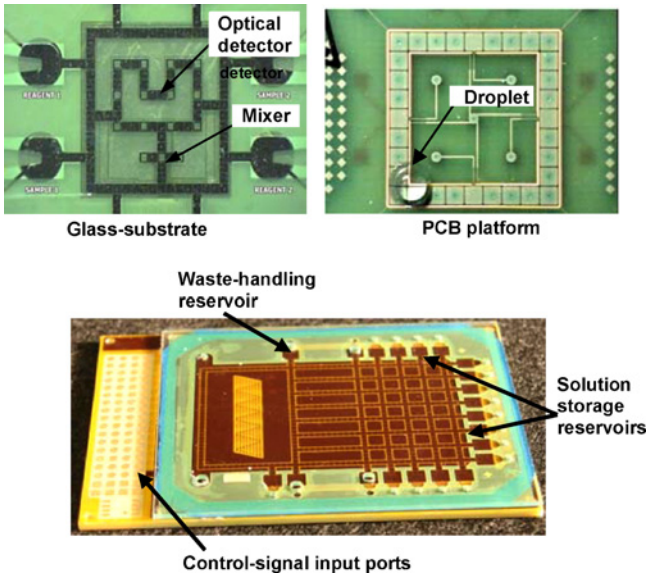


Fig. 1. Fabricated digital microfluidic arrays [26].

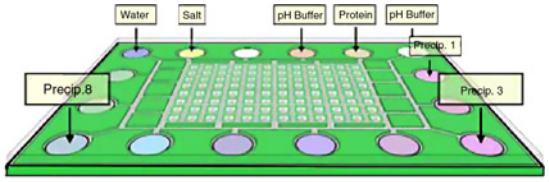


Fig. 2. Schematic view of a 96-well chip that automatically sets up 96 reagent condition solutions.

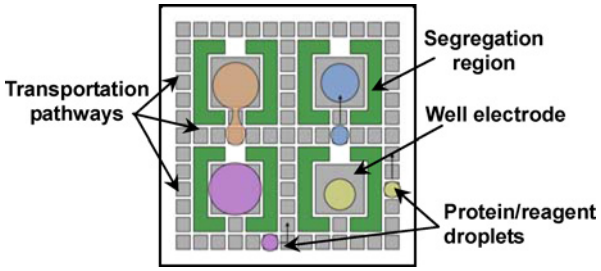


Fig. 3. Schematic top-view of four wells and the surrounding electrodes.

To address the need for low-cost, printed circuit board (PCB) technology has been employed recently to inexpensively mass-fabricate digital microfluidic biochips. Using a copper layer for the electrodes, solder mask as the insulator, and a Teflon AF coating for hydrophobicity, the microfluidic array platform can be fabricated by using an existing PCB manufacturing process [27]. This inexpensive manufacture technique allows us to build disposable PCB-based microfluidic biochips that can be easily plugged into a controller circuit board that can be programmed and powered via a standard universal serial bus port. However, a large number of independent control pins necessitates multiple PCB layers, which adds significantly to the product cost. We can address the electrodes separately by employing a serial-to-parallel interface on the device. However, this method requires active circuit components on the PCB, e.g., logic elements such as gates and flip-flops, which lead to increased cost and power consumption.

III. Multi-Well-Plate Biochip Design for Protein Crystallization

In this section, we present a multi-well-plate design prototype for protein crystallization. As discussed in Section I, to "hit" on the correct parameters for the crystallization of proteins, typically a very large number of experiments (10^3–10^4) are required. To achieve high efficiency, we use a multi-well-plate design for parallel processing, as in microbatch crystallization. The schematic for the design is shown in Fig. 2. The overall chip size is the same as that of a standard Society of Biomolecular Screening multi-well-plate. The chip has 96 wells and there are electrode pathways to connect these wells to reagent-loading and protein-loading ports.

Fig. 3 shows the specific configuration of the wells. Note that unlike microbatch crystallization, where reagents and proteins are preloaded either manually or by robotics, here reagent and protein droplets are automatically transported along the pathways from their input-loading ports to the wells. The rest of the chip real estate is used for accommodating the reagent and protein input wells. In addition to the protein reservoir that a user loads, there are two additional reservoirs that the user can load. These additional reservoirs can be loaded with any user-selected additives such as glycerol or detergents. Additives can stabilize the proteins and there are numerous reports on the use of additives to improve the quality and size of protein crystals [28]. With advances in the understanding of scaling issues, we expect designers to increase the number of wells on such a chip, since space (real estate) is available.

IV. Pin-Constrained Chip Design

Next, we assign control pins to address the electrodes in the proposed design. There are a total of 1284 electrodes in the chip, including electrodes in wells, transportation pathways, and reservoirs. If direct addressing is used, i.e., each cell of the patterned electrodes is accessed directly and independently via a dedicated control pin, a total of 1284 pins will need to be wired. However, the large number of electrodes required leads to a cumbersome wiring problem for control pins, especially when fabricated using PCB technology. In PCB technology, the diameter of the via hole is usually comparable to the electrode pitch size. Therefore, there is only a limited number of control lines that can be routed on one layer of PCB. As shown in Fig. 4, the via hole diameter is 40% of the electrode pitch. Therefore, only four control pin can be wired in any row. To route a large number of control pins, a multilayer PCB design is needed, which is prohibitively expensive. Therefore, we adopt a "correlated" pin-assignment method, which allows a control pin to be connected to multiple electrodes, thereby reducing the total number of pins.

However, the correlated pin-assignment solution introduces the problem of droplet interference. This problem can appear if multiple electrodes are controlled using a single pin. For example, assume that a droplet rests on an electrode (unit cell)

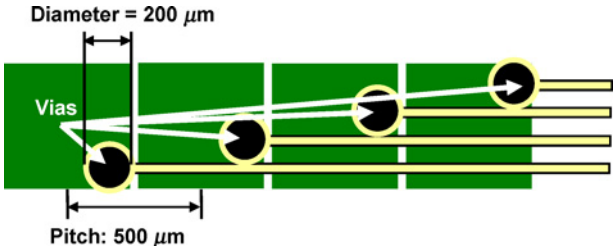


Fig. 4. Illustration of wire routing limits on a PCB layer.

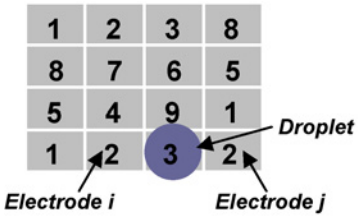


Fig. 5. Example to illustrate electrode interference.

and two of its neighbors are connected to the same pin, see Fig. 5. Recall that to move the droplet to one of two neighbors (i and j) that share the same pin, we must deactivate the electrode where the droplet rests and activate the destination electrode i. However, when electrode i is activated, the other neighbor electrode j is also activated since it shares the same pin with electrode i. In this case, the droplet undergoes a split, instead of being moved to electrode i. This problem can be solved by addressing each electrode and its neighbors with distinct pins. Since one electrode can have at most four neighbors in a 2-D array, the minimum number needed is five. Recent experimental studies have shown that five independent pins are adequate to route a droplet to any place on the chip for 1x volume manipulation [29].

A. Connect-5 Algorithm for Pin-constrained Design

Next, we address the problem of how to map control pins to the electrodes on our chip layout. An efficient and easy-to-implement algorithm is presented in this section. The algorithm (adapted from [24]) is based on a strategy of the Connect-5 (Gomoku) board game [30], thus it is referred to as the Connect-5 algorithm.

For simplicity, we first look at the application of the algorithm for a 2-D array of electrodes without wells. Our goal is to ensure that any five adjacent unit cells (a central cell and its four neighbors) that form a "cross" are assigned distinct pins. We refer to the above constraint as the "cross constraint." The pin-assignment problem under cross constraints can be mapped to the well-known vertex coloring problem in graph theory [31]. The problem here is to obtain a five-coloring of the graph derived from a biochip layout, as shown in Fig. 6. The unit cells in the microfluidic array are mapped to vertices and any two cells that belong to a "cross" are connected by an edge.

The graph coloring problem, which involves the determination of the chromatic number $\chi(G)$ for a graph G, is known to be NP-complete [31]. However, if $\chi(G)$ or the number of

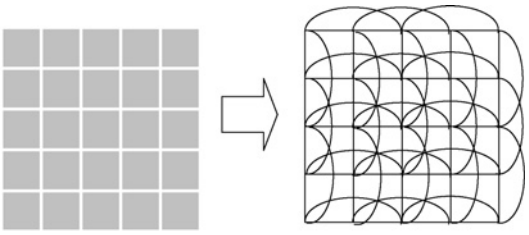


Fig. 6. Mapping of an array to an undirected graph.

colors to be used is known, as in the case here, there exist efficient algorithms for graph coloring. However, the regular structure of the arrays can be used to solve the problem more efficiently using tiling. This approach allows us to use a regular distribution of pins, a layout feature that is not directly obtained from graph coloring. The tile (or template) used here is referred to as "Bagua," a Chinese game strategy for the Connect-5 board game [30]. A Bagua is a tilted square, as shown in Fig. 7. By repeating placing Bagua structures next to each other until the array boundaries are reached, a Bagua repetition is derived as shown in Fig. 8. The tiling using Bagua repetitions forms the basis for the Connect-5 algorithm.

Five copies of Bagua repetitions are sufficient to cover a biochip array of any size. This is because of the following property of a Bagua repetition: vertices connected to the same (shared) pin appear after exactly five cells in the same row or column of the array. The array can be covered with Bagua repetitions by simply taking a Bagua repetition and shifting it one cell along an arbitrary direction, e.g., upward, then assigning it to another control pin and repeating this step four times, as shown in Fig. 9. Note that, although the direction of shift is arbitrarily selected at the start of the tiling process, once chosen it must be consistent over the four shifting steps. In other words, the shift direction, once chosen, must remain fixed during the assignment

Next, we show that control pins assigned to the electrodes using this method in a microfluidic array allow free movement of a single droplet, i.e., the "cross constraint" is met. To demonstrate this, we consider the cell that is hatched in Fig. 9. If the cell is assigned Pin 1, we cannot assign the same pin to the unit cells that are shaded. Otherwise, we will violate the cross constraint in some cases. It can be found that all the unit cells in the Bagua tile and its repetitions stay out of the forbidden area. Thus, for each pin assigned to cells in a Bagua repetition, the cross constraint is not violated. Since this is true for any Bagua repetitions and any array can be tiled by five copies of Bagua repetitions, the "cross constraint" is automatically met for every cell in our pin-assignment method.

B. Modified Connect-5 Pin-assignment Algorithm for Irregular Chip Layout

The Connect-5 algorithm described in Section IV-A assumes that the chip has a regular layout, i.e., a rectangular array. However, in practical chip design, to reduce production cost, unused electrodes are often removed from the rectangular array, resulting in an irregular chip layout. For example, in the multi-well-plate chip design shown in Fig. 3, several electrodes

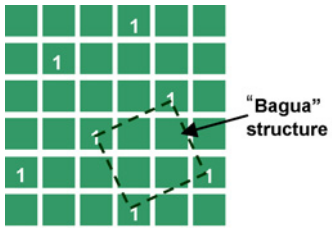


Fig. 7. Single Bagua structure (the tilted square) and its repetition in a square microfluidic array.

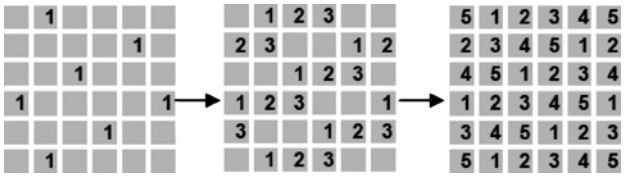


Fig. 8. Covering a microfluidic array by shifting Bagua repetition along rows.

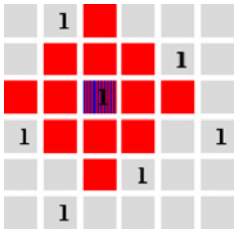


Fig. 9. Demonstration that the "cross constraints" are met.

have been removed. The avoid space left is used as segregation regions.

Here, we modify the Connect-5 pin-assignment procedure to make it applicable for such chip design with irregular chip layout, more specifically, for our well-plate design. Note that the well-plate design can be viewed as a special case of the 2-D array where parts of the array are occupied by wells and segregation walls. Unoccupied electrodes between wells can be used as transportation pathways. Therefore, the pin-assignment for these electrodes does not need to be changed. The overall pin-assignment procedure is as follows.

1) Start with a 2-D electrode array of the same size as the target well-plate design, but with no cells reserved as wells or segregation regions.
2) Apply the Connect-5 algorithm to generate a preliminary pin-assignment result. For example, to generate a pin-assignment result to the multi-well chip in Fig. 3, a preliminary result is first derived, as shown in Fig. 10(a). Note that since the pin-assignment can be carried out by copying and shifting of the bagua repetition, it takes $O(N^2)$ time for an $N \times N$ array.
3) Next, consider the electrodes that will make up the segregation regions and wells in the multi-well design. Disconnect these electrodes from their control pins, see Fig. 10(b). This step also takes $O(N^2)$ time for an $N \times N$ array.

Finally, group the electrodes occupied by each well and connect each group to a single control pin. For independent control of each well, the group control pins must be different not only from each other but also from the pins assigned to the electrodes on the transportation pathway. The modified pin-assignment result is shown in Fig. 10(c). This step takes $O(N^2)$ time, which makes the computation complexity of the entire pin-assignment algorithm $O(N^2)$.

Note that in Fig. 10(c), the same patterns of pin assignment repeat in both column and row directions with a period of 6. Based on this observation, we can adjust the size of the unit well to obtain a more regular pin-assignment result. Here, define a *well unit* as a single well and the routing pathways round it. In the design in Fig. 10(c), the size of the well unit is 7×7. We first shrink the size of the unit well from 7×7 to 6×6 (since the period of the repetitive pin-assignment patterns is 6) electrodes, as shown in Fig. 11. Next, we apply the Connect-5 algorithm to get a pin assignment for the 96-well chip with the adjusted unit well size, see Fig. 12.

For a 96 well-plate design with well unit size of 5×5, there are a total of 1284 electrodes in the chip, including 96 electrodes in wells, 1108 electrodes composing of transportation pathways and 80 electrodes serving as dispensing electrodes and storage reservoirs. Therefore, a total of 1284 control pins are needed for direct addressing. In contrast, the design in Fig. 12 only needs five pins to control all the electrodes on the transportation pathways, thereby significantly reducing the total number of control pins to $5 + 96 + 80 = 181$.

The pin-constrained design that uses the Connect-5 method not only significantly reduces the number of control pins but it also provides an easy wiring solution. According to [24], electrodes sharing the same pin in the pin-assignment result from Connect-5 algorithm are diagonally aligned. Therefore, they can be easily wired diagonally, as shown in Fig. 13.

Moreover, the diagonal wiring allows the diameter to be almost the same as the electrode pitch size, as illustrated in Fig. 14. This efficient wiring plan allows the 181 pins to be wired on a two-layer PCB. Recall that the direct-addressing method needs 1284 control pins, which requires a four-layer PCB and thereby increases the fabrication cost by a factor of 1.6–2 [32]. Moreover, the 181 pins can be easily incorporated using standardized 3 mil feature size technology. In contrast, to fit the 1284 pins in the direct-addressing-based design, 2 mil technology, which usually cost 3–5x times more than 3 mil technology, has to be used. Therefore, the pin-constrained design achieves a reduction in fabrication cost by a factor of 5–10x. The reduction is more significant when the wiring-plan design cost is considered.

In Fig. 12, every well unit has the same pattern of pin-assignment. This is because the dimension of the unit well is the same as the period of pin-assignment patterns from Connect-5 algorithm. This regular pin-assignment result facilitates the use of an efficient well-loading algorithm, discussed in Section V.

C. Pin-Assignment for Manipulation of 2x Droplets

The previous discussion in this section is based on an assumption that all the droplets to be routed are of unit volume, which is defined as the standard volume of a unit droplet dispensed from the source reservoir. We refer to these droplets

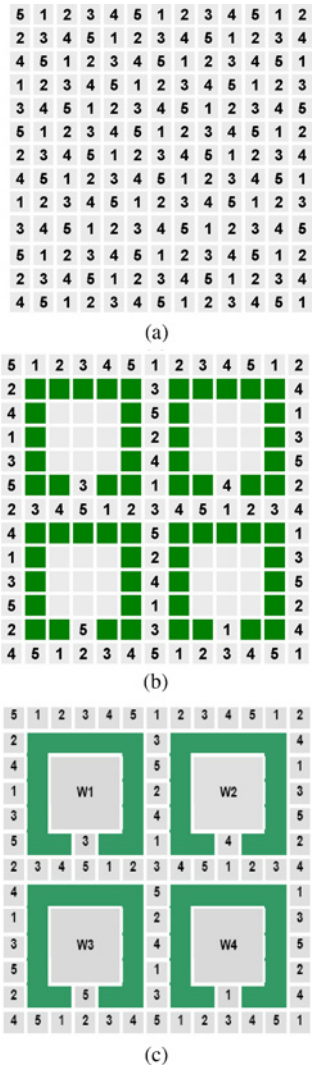


Fig. 10. Example of pin-assignment example for a 4-well-plate design.

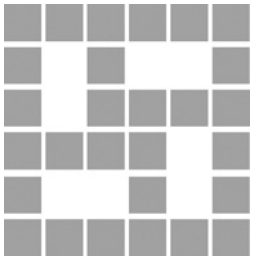


Fig. 11. Illustration of a 6 × 6 electrode well unit.

as 1x droplets. However, in many large scale applications including protein crystallization, in order to reduce droplet routing time, sometimes we need to route droplets of larger volume. A common practice is to dispense two droplets of unit size from the source reservoir and merge them to form a larger droplet, referred to here as a 2x droplet. The manipulation of 2x droplets is similar to the manipulation of 1x droplet. The difference lies in the fact that to transport/hold a 2x droplet, instead of activating/deactivating a single electrode, we need

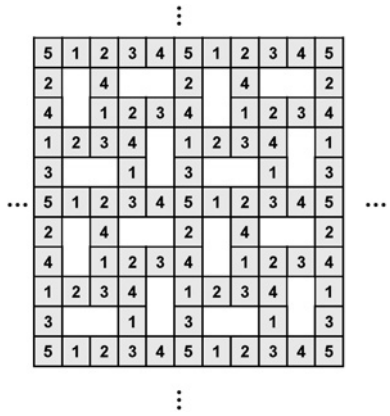


Fig. 12. Pin assignment using five pins for the 96-well chip (unit well size = 6 × 6 electrodes).

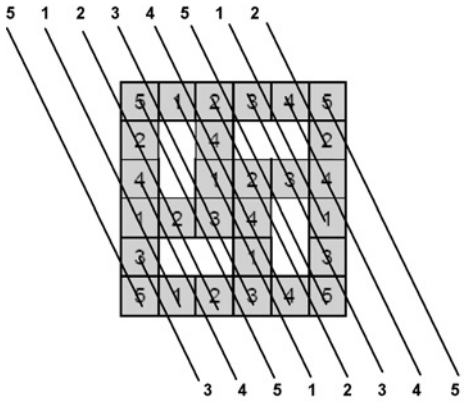


Fig. 13. Wiring of a well unit.

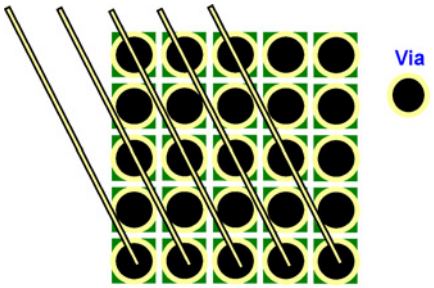


Fig. 14. Wiring of a well unit with large vias.

to activate/deactivate a pair of two adjacent electrodes as a group, as shown in Fig. 15.

Such 2x droplet manipulations can help reduce the droplet routing time significantly. As an example, suppose that we need to dilute a stock solution of Reagent A with concentration C to prepare a target solution of concentration $C/5$. Assuming only manipulation of 1x droplets is allowed, we have to route one droplet from the stock-solution reservoir and four droplets from the diluents reservoir. A total of five routing iterations are needed. Using 2x droplet transportation, we can route one 1x droplet from stock solution and two 2x droplets from diluents solution. As a result, only three routing steps are needed.

Next, we investigate the feasibility of carrying out 2x droplet manipulation on the Connect-5 algorithm-based

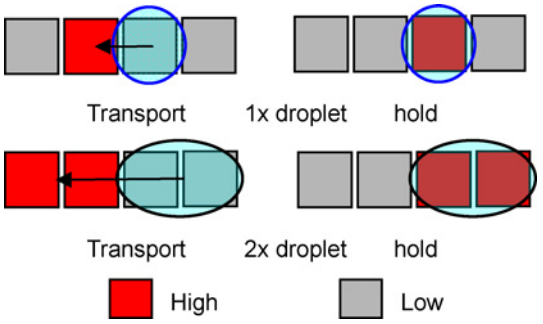


Fig. 15. Comparison of electrode activations between 1x and 2x droplet manipulation.

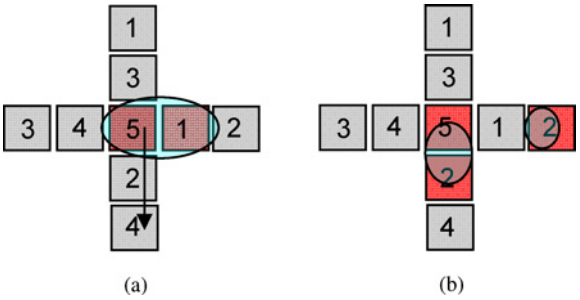


Fig. 16. Example of electrode interference resulting from the manipulation of a 2x droplet on the five-pin assignment layout.

pin-constrained design described in Section IV-A. Unfortunately, the manipulation of 2x droplets on the proposed pin-constrained design may suffer from electrode interference. An example is shown in Fig. 16. At first, a 2x droplet is seated on the two electrodes in the center of the array, as shown in Fig. 16(a). In the next clock cycle, we want to move the 2x droplet one electrode downward. To do this, we need to activate Pin 5 and Pin 2. However, note that in Fig. 16(a), the electrode on the right of the 2x droplet is also connected to Pin 2, therefore, it will also be activated. In this case, the droplet undergoes a split, instead of being moved downward, as shown in Fig. 16(b).

Next, we propose a modified pin-assignment method to support 2x droplet manipulation without electrode interference. Recall that 2x droplet manipulation is the same as 1x droplet manipulation except that in 2x droplet manipulation, electrodes need to be activated and deactivated in pairs. Therefore, if we treat a pair of adjacent electrodes as a large unit electrode, 2x droplet manipulation will be exactly like 1x droplet manipulation, just on a larger scale (a scale where a unit electrode is of 2x volume, here referred to as 2x scale).

Following this observation, we can derive a pin-assignment result that supports 2x droplet manipulation based on the Connect-5 algorithm-based pin-assignment layout. The procedure is list as follows. For better explanation, we use an example to illustrate the details for each step of the procedure.

1) Start from the Connect-5 algorithm-based pin-assignment layout; see the example in Fig. 17(a).
2) Substitute each electrode in the pin-assignment result with two smaller electrodes and substitute each pin with a set of two independent pins. Two rules apply.

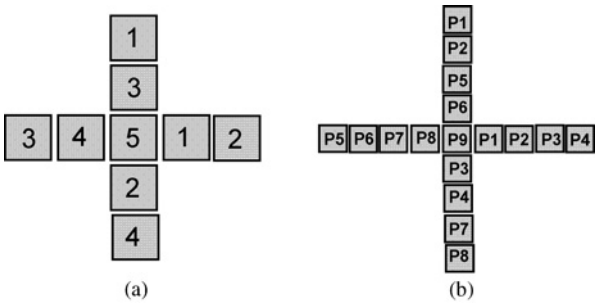


Fig. 17. Illustration of the procedure to generate pin-assignment supporting 2x droplet manipulation via electrode and pin substitution.

   a) The pin substitution must be consistent, i.e., one pin can only be substituted by a fixed set of pins.
   b) The pin set substituting different pins must be mutually exclusive.

   As shown in Fig. 17(b), the pin substitution mapping is $\{1\} \rightarrow \{P1, P2\}, \{2\} \rightarrow \{P3, P4\}, \{3\} \rightarrow \{P5, P6\}, \{4\} \rightarrow \{P7, P8\}$.
3) Note that the electrode at the intersections in the array can be substituted by a set of two electrodes aligned in either column or row directions. To eliminate randomness in such substitution, we do not substitute the electrodes in the intersections and assign only a single independent control pin to them. As shown in Fig. 17(b), electrodes in the intersections were assigned Pin 5, which is now substituted by pin P9.

The resulted new pin-assignment layout can support 2x droplet manipulation. This is because the electrode and pin substitutions carried out in the above procedure do not change the structure of the layout and the relations between electrodes and pins. Independent pins in the pin-assignment before the substitutions are replaced by mutually exclusive pin sets; electrodes sharing the same pins are still sharing the same pin sets after the substitution. Therefore, moving a 2x droplet on the pin-assignment layout after the substitution is the same as moving a 1x droplet on the original pin-assignment layout, which is electrode-interference-free. As a tradeoff, a chip of 2x the original size is needed.

The modified pin-assignment based on electrode and pin substitutions can be easily adopted to support the manipulation of droplets of larger volume. The only change is the number of pins in the pin set for substitution. Note that in the substitution, only four pins can be substituted by pin sets. The 5th pin is for intersection electrode thereby can only be mapped to another single pin. Therefore, we can drive the formula of the number of control pins required to manipulation nx droplet, i.e., $4n + 1$. We use this formula to calculate the number of control pins required for manipulation of multi-x droplet, as listed in Table I.

D. Consideration of Diagonal Electrode Interference

The pin-assignment algorithm described in Section IV-A, and Section IV-B also assumes that the droplet cannot be affected by the activation of "diagonal" electrodes. However, droplet of larger volume may undergo diagonal movement because of overlapping with diagonal electrodes; see Fig. 18.

TABLE I
NUMBER OF CONTROL PINS REQUIRED FOR MANIPULATION OF nX DROPLET (n = 1, 2, 3, 4)

	1x	2x	3x	4x
# of control pin	5	$4 \times 2 + 1 = 9$	$4 \times 3 + 1 = 13$	$4 \times 4 + 1 = 17$

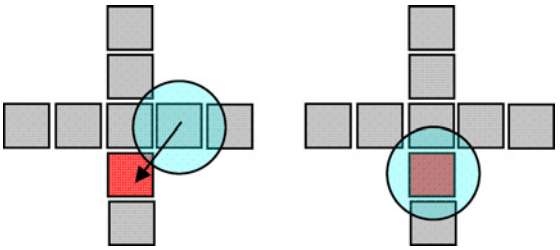


Fig. 18. Illustration of diagonal droplet movement.

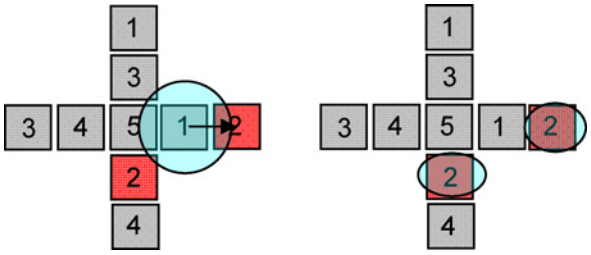


Fig. 19. Example of electrode interference resulted from diagonal droplet movements.

In this case, pin-assignment procedure proposed in Section IV-A, and Section IV-B cannot be used due to electrode interference. A example is shown in Fig. 19. To move the droplet one electrode to the right, we need to activate Pin 2. This will also activate the diagonally adjacent electrode. As a result, the droplet undergoes a split.

The diagonal electrode interference problem can also be solved by using the pin-assignment result generated in Section IV-C. Note that the diagonal electrode interference cannot happen at intersection electrodes because these electrodes have no diagonally adjacent neighbors. For a nonintersection electrode λ_1, diagonal electrode interferences may occur if any two of its neighbors (λ_2 and λ_3) are sharing the same control pin, as shown in Fig. 19. However, note that such nonintersection electrode (λ_1) will be replaced with a pair of two electrodes in the substitution procedure described in Section IV-B. For either of two smaller electrodes, the other one will serve as a guard electrode to keep one of its two old neighbors (λ_2 or λ_3) that were causing diagonal electrode interference at least two electrode away, as shown in Fig. 20, so that no diagonal electrode interference is possible.

V. SHUTTLE--PASSENGER-LIKE WELL-LOADING ALGORITHM

In this section, we focus on the problem of loading the wells with sample and reagent droplets on the pin-constrained chip. For simplicity, we focus our discussion on well-loading with using 1x volume droplet.

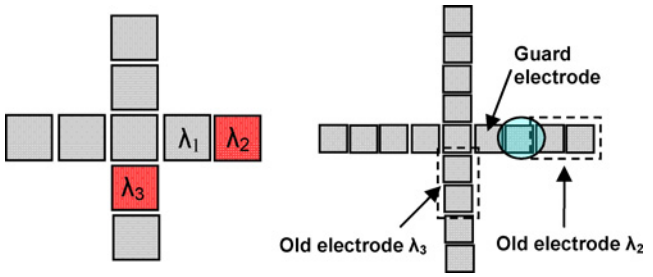


Fig. 20. Illustration of avoiding diagonal electrode interference using electrode/pin substitutions.

A. Manipulation of Sample and Reagent Droplets

The goal is to efficiently route the sample and reagent droplets to their destination wells. Note that in the 96-well chip design in Fig. 12, every 6×6 well unit has the same pattern of pin-assignment. Therefore, any sequence of manipulations in a single well unit will cause the same manipulations in all the other well units. Although this "synchronizing" property leads to reduced freedom of droplet manipulations, it allows the concurrent manipulation of multiple droplets. Based on this observation, we propose a parallel shuttle-passenger-like within the same unit, just as synchronized shuttles that carry passengers from fixed start points to fixed destinations. The shuttles run regularly irrespective of whether there is any passenger. To go to a specific destination, a passenger needs to get to the correct starting point and wait for the shuttle (pin actuation sequence) for pick-up and routing to the destination (well).

Routing of droplets to the starting point can also be carried out using the shuttle-passenger-like method. As in the example in Fig. 21, the routing step can be carried out using the shuttle (pin activation sequence) as shown in Fig. 22.

Therefore, the proposed well-loading method contains two steps. In the first step, droplets to be routed are transported to the corresponding start points in their destination well units. This step is carried out as follows.

1) Calculate the electrode activation sequence to route the droplet to the starting point farthest away from the source reservoir. This step takes O(N) time for an $N \times N$ array.
2) Select a subsequence from the sequence from Step 1 for each droplet that can be routed to its starting point. Assume the selection is carried out using brute-force searching, this step takes O(NM) time, where M is the number of droplets. Note that in the shuttle-passenger-like well-loading algorithm, the number of droplets is no more than the number of well units, i.e., $M < N^2$. Therefore, this selection step takes O(N^3) time.
3) Applying the electrode-activation sequence from Step 1, and dispense each droplet at a specific time corresponding to the start of its subsequence. This step takes O(N) time, which makes the computation complexity of the entire well-loading method O(N^3) for an $N \times N$ array.

Next, a second pin-actuation sequence is applied to route droplets to their target wells. The overall routing steps take little time because all the wells can be filled using only two pin-actuation sequences.

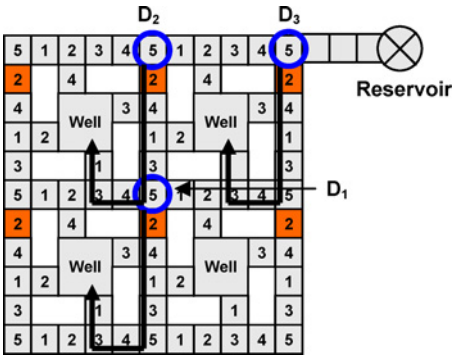


Fig. 21. Loading of three droplets using the shuttle-passenger-like method.

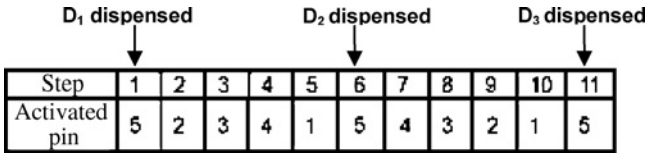


Fig. 22. Activation sequence and dispensing time instances for routing droplets to corresponding starting point Fig. 21.

B. Manipulation of Wash Droplets

Excessive routing of protein droplets may result in the problem of protein adsorption. This is because proteins tend to adsorb irreversibly to hydrophobic surfaces and contaminate them. Silicone oil with its low-surface tension and spreading property has been advocated as a filler medium for protein assays to prevent contamination [14]. However, it has also been reported that the use of silicone oil alone is not sufficient for many types of proteins [33]. A set of wash droplets is typically used in such cases for surface cleaning after several iterations of parallel well-loading process, especially for unit cells that are shared by droplet routes.

Note that, the manipulation of the wash droplets is the same as the manipulation of sample and reagent droplets. Therefore, we can use the same shuttle-passenger-like well-loading algorithm to route wash droplets for surface cleaning. This shuttle-passenger-like well-loading algorithm can also be used to manipulate 2x droplets concurrently.

VI. Chip-Testing

To ensure reliability, the proposed chip design needs to be tested thoroughly. We only focus here on structural test, where the goal is to route test droplet to traverse the target array for defect detection. In this section, we discuss the adaptation of a recently proposed structural test method refer to as parallel scan-like test to the proposed chip design [34].

A. Parallel Scan-like Test

Parallel scan-like test includes a cost-effective fault detection and a rapid diagnosis method based on test outcomes. Given a microfluidic array, scan-like test is carried out the in parallel using multiple droplets. Each column/row in the array is associated with a test droplet and its "target region." A target region for a droplet includes the cells that are traversed by this droplet as shown in Fig. 23.

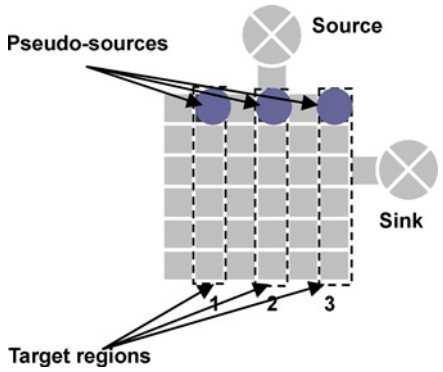


Fig. 23. Example of target regions and pseudo-sources.

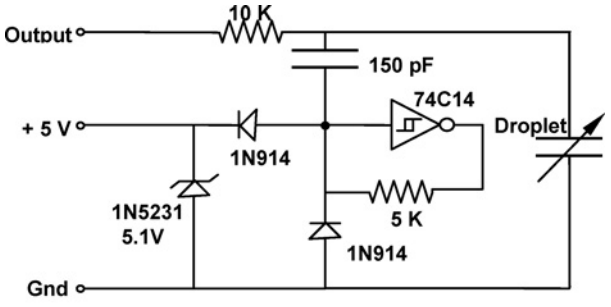
Test droplets are dispensed from the test droplet source to the start electrodes of their target regions (columns/rows), as shown in Fig. 24. For each target region, the start electrode acts as the test-droplet source for the underlying single-droplet scan-like method. They are referred to as *pseudo-sources*. Starting from these pseudo-sources, test droplets are routed in parallel to the electrodes at the other end of the corresponding target regions. These end-points are referred to as pseudo-sinks. Finally, the test droplets are routed to the sink reservoir. Dispensed from the single source, test droplets are aligned one-by-one and routed in sequence, like components in an assembly line, along the periphery nodes to their pseudo sources. The reverse process is carried out when the test droplets are routed from the pseudo sink to the sink reservoir.

The test outcome is read out using a capacitive sensing circuit located at the sink reservoir [34], [35], as shown in Fig. 24. In Fig. 24, a capacitance controlled oscillator is connected to the electrode at the detection site. The capacitance between the control electrode and the ground plane modulates the frequency of the ring-oscillator. The frequency was measured using a counter on the data acquisition board. When a droplet is routed to transverse the electrode at the detection site, its motion will incur a pulse-like fluctuation of the capacitance value in the oscillator circuit, which in turn results in a fluctuation in the frequency the oscillator. Such a fluctuation is reflected at the output of the counter as a pulse.

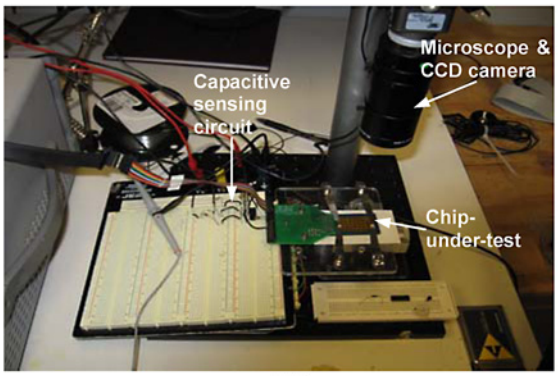
The capacitive sensing circuit can generate a pulse sequence corresponding to multiple test droplets. Different fault patterns (i.e., groups of failing cells) are mapped to different pulse sequences. Consider the example shown in Fig. 25. The output pulse sequence indicates a defect in the fifth column. The defect site can be precisely identified by carrying out another round of such parallel scan-like test in the row direction.

The complete parallel scan-like test procedure is as follows.

1) *Step 1 (Peripheral Test):* A test droplet is dispensed from the source. It is routed to traverse all the electrodes on the boundary of the microfluidic array, referred to as *peripheral electrodes*, and the droplet finally returns to the sink.
2) *Step 2 (Column Test):* Two iterations of parallel scan-like test with one column shift are carried out. This step



(a)



(b)

Fig. 24. Capacitive sensing circuit. (a) Circuit design [34], [35]. (b) Experimental setup.

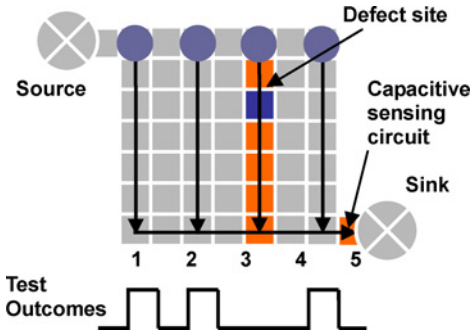


Fig. 25. Example of test outcome for a faulty array.

tests every single cell in each column. Therefore, it is referred to as "column test."

3) *Step 3 (Row Test):* Repeat parallel scan-like test (two iterations) for the rows to detect defects involving pairs of adjacent cells in each row. This step is referred to as "row test."

B. Application to Multi-Well-Plate Pin-Constrained Chip

Next, we adapt the parallel scan-like method to the well-plate design. As mentioned in Section IV-A, our well-plate design can be viewed as a special case of a 2-D array where parts of the array are occupied by wells and segregation walls. Unoccupied electrodes between wells are used as transportation pathways.

Here, we focus on the testing of the transportation pathway, since defects in the transportation pathway will not only block well-loading, but also affect the droplet routing among different well units. Note that the transportation pathways are in fact composed of columns/rows of electrodes. Therefore, we should be able to test these transportation pathways in parallel in the same way as the columns and row test in the proposed parallel scan-like test procedure, as shown in Fig. 26.

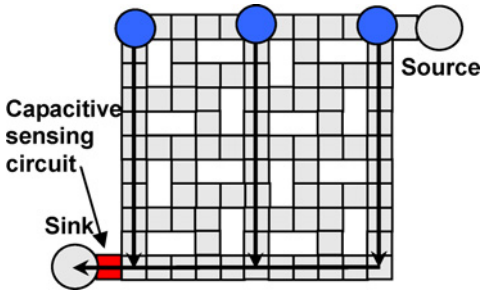


Fig. 26. Parallel scan-like test on multi-well chip.

As shown in Fig. 26, multiple test droplets are routed in parallel to test the target transportation pathways separated by five columns (the length of a well unit, minus one, equals five). In this way, all the transportation pathways can be covered using one round of column test and one round of row test.

Next, we discuss how the above test scheme can be applied when pin-constrained design is used. Here, we carry out step by step analysis to the test procedure described in Section VI-A. For each step in the procedure, we check whether the droplets movement patterns required in that step are allowed by the pin-assignment layout.

The first step, i.e., peripheral test, involves the routing of a single test droplet to traverse all the peripheral electrodes. As we mentioned in Section IV-A, pin-assignment result generated using Connect-5 algorithm allows free movement of a single droplet. Therefore, this peripheral test step can be easily carried out on a pin-constrained chip.

The second step, i.e., column test, requires synchronous routing of multiple test droplets to traverse every single cell in each column of the microfluidic array. Note that here in the multi-well chip design, not every column of the array is used and thereby needs to be tested. In fact, it is only every 5th column that is used as a routing pathway. Therefore, only a subset of the column test needs to be carried out, as shown in Fig. 27. Test droplets are routed to the pseudo sources at the beginning of every 5th column. Next, these test droplets are moved in parallel to traverse the corresponding column and finally routed in sequence to the sink reservoir for error detection.

This column test step requires synchronous movement of multiple test droplets while maintaining five-column distance. As we discussed in Section V while introducing the shuttle-passenger-like well-loading algorithm, such synchronous movements are allowed by the Connect-5 pin-assignment result.

The final step, i.e., row test, can be carried out in the same way as the column test.

As discussed in Section VI-A, the test outcome is read out using a capacitive sensing circuit located at the sink reservoir. Different fault patterns are mapped to different pulse

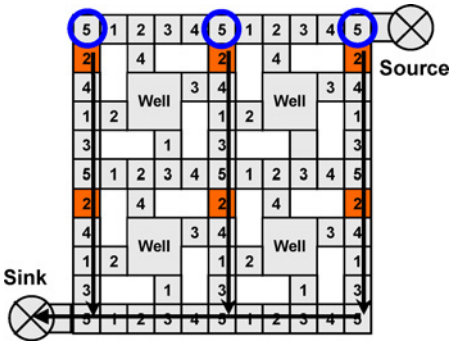


Fig. 27. Parallel scan-like test on multi-well, pin-constrained chip.

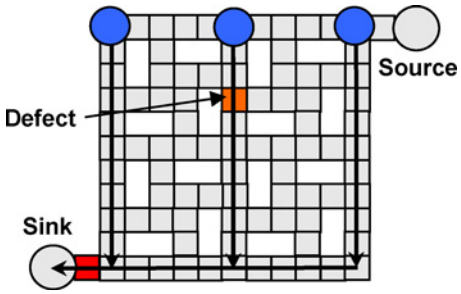


Fig. 28. Example of a defect that cannot be precisely localized by column and row test.

sequences observed at the capacitive sensing circuit. However, note that here defects in the multi-well design cannot be precisely localized by carrying out column test and row test together. This is because in the multi-well chip design, not all the columns and rows are used as transportation pathways. There are many electrodes on the transportation which are not at the intersections of any used column and rows, as shown in Fig. 28.

In the case shown in Fig. 28, the sensor readout indicates an error for only one step, i.e., column-test. As a result, we can only determine which column the defect resides on based on the column and row test outcome. As shown in Fig. 29, we divide the defective column/row into two partitions and route test droplet to transverse each of them. Based on test readout, we can determine which partition has the defective cell, thereby cutting the search space in half. Such a partitioning-and-testing procedure is carried out iteratively until the exact location of the defective cell is determined.

VII. Defect Tolerance

Using the testing and diagnosis methods proposed in the previous section, we can easily locate defect sites on the chip. In this section, we propose a "cross loading" based method to achieve defect tolerance for the proposed chip design.

We first classify defects into three categories based on their locations on the chip. Note that the well-loading algorithm proposed in Section V, wells are loaded from one side, i.e., right side or left side. Therefore, not all the wells are loaded from the right/left side. Therefore, not all electrodes are used. If a defect occurs in these unused electrodes, then it will not affect droplet manipulations on the chip. We refer to this

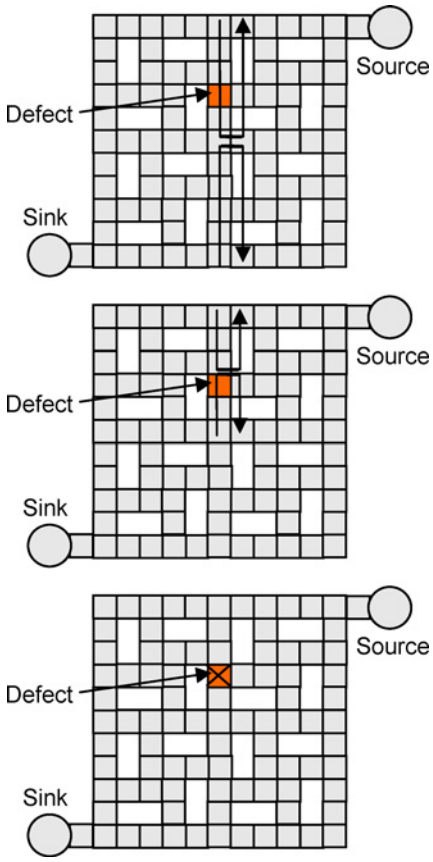


Fig. 29. Diagnosis of a defect on a nonintersection electrode.

type of defects as *benign defects*. In the design proposed in Section IV-A, benign defects include all the defects in the unused entrance electrodes for the well and all the electrodes between the bottom entrance electrodes and the left/right routing pathways. For benign defects, no defect tolerance is needed.

The second category of defects occurs on the electrodes used by the well-loading algorithm on the electrode rows but not on the routing pathways. These defects are referred to as *loading pathway defects*, as shown in Fig. 30. These defects can be bypassed by simply changing the side from which the well is loaded.

The third category includes all the defects on the routing pathways. Therefore, we refer to them as *routing pathway defects*. Unlike loading pathway defects, these defects affect the loading operations for more than one well unit. They cannot be bypassed by simply changing the side from which the well is loaded. Instead, we use a "cross loading" method for defect tolerance. Two iterations of well-loading operations are carried out, one in the column direction and one in the row direction. If the defects occur on the routing pathways in the well-loading operation in the column direction, the loading of all the wells within the same column with the defects will be skipped. The skipped wells will then be loaded in the well-loading operation in the row direction and vice versa. An example is shown in Fig. 31. There is a defect in the routing pathway in the first well unit in the second column.

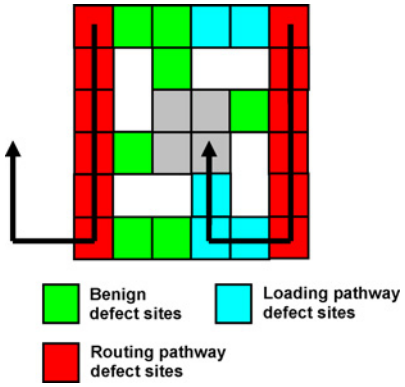


Fig. 30. Three categories of defects.

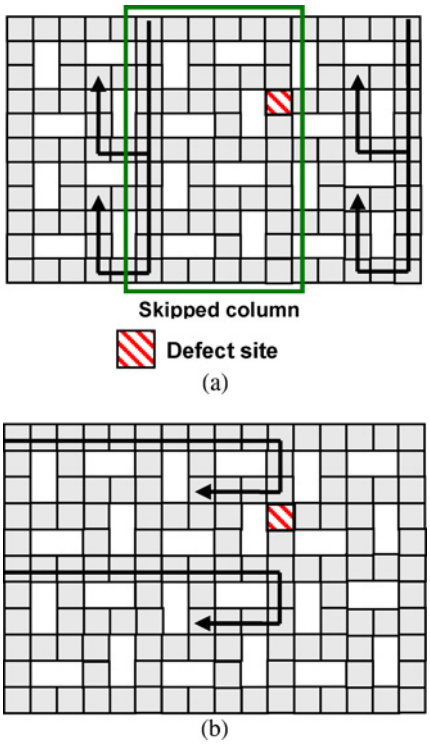


Fig. 31. Illustration of "cross loading" method. (a) Loading in column direction. (b) Loading in row direction.

In the column-loading step, the entire second row is skipped. The skipped wells are then loaded in the second iteration of loading in the row direction.

VIII. Evaluation of Well-Loading Algorithm and Defect Tolerance

In the section, we evaluate the proposed pin-constraint design and the shuttle-passenger-like well-loading algorithm.

A. Loading Time

We first calculate the time needed for loading the wells on a pin-constrained chip and a chip with independent pins (direct-access). In a direct-access chip, the time required to load all the wells is determined by the time taken by a droplet to traverse the critical path, i.e., from the dispensing reservoir

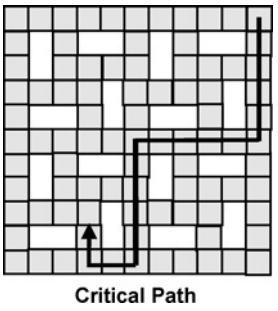


Fig. 32. Critical path for the multi-well chip (for both the direct-access and pin-constrained chips).

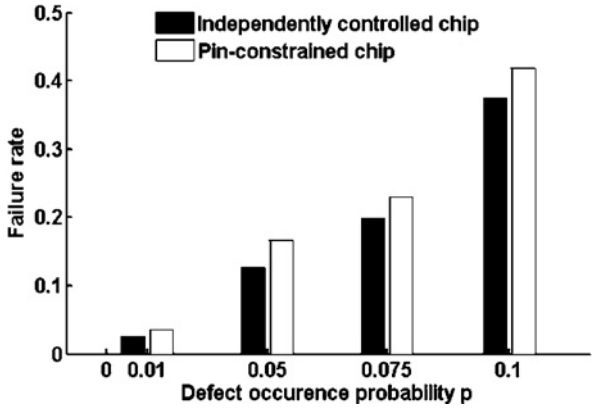


Fig. 33. Evaluation of failure rates for pin-constrained chip and independently controlled chip.

to the farthest well, as shown in Fig. 32. For an $N \times N$ array, the routing time for the critical path is $2N - 3$ clock cycles. The proposed pin-constrained chip has the same critical path. Using the well-loading algorithm from Section V, a droplet can be routed along the critical path one electrode per clock cycle with no stalled cycles. Therefore, the routing time is also $2N - 3$ clock cycles. Thus, we conclude that the pin-constrained design provides the same routing efficiency as the direct-access design, while it achieves a significant reduction in the number of control pins.

B. Defect Tolerance

Next, we examine the defect tolerance of the proposed pin-constrained design by injecting random defects. A design is deemed to be robust if the injected defect can be bypassed using the defect-tolerance methods proposed in Section VII. Some defects may block all the routing pathways to one or more wells, and these wells cannot be loaded. In this case, a failure occurs on the chip.

Next, we define a parameter referred to as "failure rate." Let N_t be the total number of biochips in a representative sample, and let N_f be the number of defective chips that suffers from a failure. Then the *failure rate* f is defined by the equation $f = N_f/N_t$.

We run the simulations with difference defect occurrence probabilities for the pin-constrained chip and record the failure rates. As a baseline, we also carry out defect injection for a direct-access chip. Results are obtained by averaging outcomes

from 1000 simulation runs, see Fig. 33. Note that if we do not set any upper limit on the well-loading time, any defect that can be bypassed in the direct-access chip can also be bypassed in the pin-constrained chip. This is because we can manipulate only one droplet to load only one well in any iteration of shuttle-passenger-like routing, which allows the same degree of freedom as in the direct-access chip. However, this scheme results in a significant increase in the well-loading time. Therefore, in our evaluation, we use a restricted definition of failure for the pin-constrained design; it refers to the case that the injected defects cannot be bypassed using the "cross loading" method.

Fig. 33 shows that, as expected, the introduction of pin constraints leads to a slightly higher failure rate compared to the direct-access chip. However, this increase is acceptable in practice due to the significant reduction in the number of control pins for the proposed design.

IX. Conclusion

We have presented a multi-well-plate based digital microfluidic biochip design for protein crystallization. The proposed biochip is capable of concurrently setting up 96 conditions, thereby achieves high-throughput. We have also applied an efficient algorithm to generate a pin-assignment plan for the proposed design, which enables control of the biochip with only a small number of pins. Compared to directly addressable biochip, the proposed pin-constrained design achieves a significant reduction in fabrication cost. We have also described efficient droplet-routing algorithms for defect-tolerant well-loading. The propose chip design and associated routing algorithms will pave the way for increased use of digital microfluidic biochips in high-throughput, highly automated, and affordable protein crystallization systems. This paper has, therefore, addressed the need for research on functional diversification in electronic systems ("Higher Value Systems" and "More than Moore"), as described in the 2007 International Technology Roadmap for Semiconductors document.

References

[1] A. Fersht, *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. New York: Freeman, 1998.

[2] J. F. Kennedy and C. Bandaiphet, *Protein Structure: Determination, Analysis, and Applications for Drug Discovery*. New York: Marcel Dekker, 2003.

[3] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips, "A 3-D model of the myoglobin molecule obtained by X-ray analysis," *Nature*, vol. 181, no. 4610, pp. 662–666, 1958.

[4] A. McPherson, "Crystallization of macromolecules: General principles," *Methods Enzymol. A*, vol. 114, pp. 112–120, 1985.

[5] N. E. Chayen, "Recent advances in methodology for the crystallization of biological macromolecules," *J. Crystal Growth*, vol. 198, no. 199, pp. 649–655, 1999.

[6] N. E. Chayen, P. D. S. Stewart, D. L. Maeder, and D. M. Blow, "An automated system for micro-batch protein crystallization and screening," *J. Appl. Crystallogr.*, vol. 23, no. 4, pp. 297–302, 1990.

[7] J. R. Luft, D. M. Rak, and G. T. DeTitta, "Microbatch macromolecular crystallization in micropipettes," *J. Crystal Growth,* vol. 196, nos. 2–4, pp. 450–455, 1999.

[8] http://www.douglas.co.uk/oryx8.htm

[9] http://www.gilson.com/Resources/A_Compact_Automated_Liquid_Handler_ For_Microbatch_Protein_Crystallography.pdf

[10] http://www.syrrx.com/technology/index.htm

[11] R. C. Stevens, "High-throughput protein crystallization," *Current Opinion Struct. Biol.*, vol. 10, no. 5, pp. 558–563, Oct. 2000.

[12] H. I. Krupka, B. Rupp, B. W. Segelke, T. P. Lekin, D. Wright, H. C. Wu, P. Todd, and A. Azarani, "The high-speed hydra-plus-one system for automated high-throughput protein crystallography," *Acta Crystallogr.*, vol. D58, pp. 1523–1526, Sep. 2002.

[13] R. B. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible?," *Microfluidics Nanofluidics*, vol. 3, no. 3, pp. 245–281, 2007.

[14] R. B. Fair, A. Khlystov, T. D. Tailor, V. Ivanov, R. D. Evans, P. B. Griffin, V. Srinivasan, V. K. Pamula, M. G. Pollack, and J. Zhou, "Chemical and biological applications of digital-microfluidic devices," *IEEE Design Test Comput.*, vol. 24, no. 1, pp. 10–24, Jan.–Feb. 2007.

[15] K. F. Böhringer, "Modeling and controlling parallel tasks in droplet-based microfluidic systems," *IEEE Trans. Comput.-Aided Design Integr. Circuit Syst.*, vol. 25, no. 2, pp. 329–339, Feb. 2006.

[16] E. J. Griffith, S. Akella, and M. K. Goldberg, "Performance characterization of a reconfigurable planar array digital microfluidic system," *IEEE Trans. Comput.-Aided Design Integr. Circuit Syst.*, vol. 25, no. 2, pp. 340–352, Feb. 2006.

[17] K. Chakrabarty and F. Su, *Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques*. Boca Raton, FL: CRC Press, 2006.

[18] T. Xu and K. Chakrabarty, "Integrated droplet routing in the synthesis of microfluidic biochips," in *Proc. IEEE/Assoc. Comput. Machinery Design Automat. Conf.*, 2007, pp. 948–953.

[19] M. Cho sand D. Z. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1714–1724, Oct. 2008.

[20] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "BioRoute: A network-flow based routing algorithm for digital microfluidic biochips," in *Proc. Int. Conf. Comput.-Aided Design*, 2007, pp. 752–757.

[21] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. "Placement of defect-tolerant digital microfluidic biochips using the T-tree formulation," *Assoc. Comput. Machinery J. Emerging Technol. Comput. Syst.*, vol. 3, no. 3, Nov. 2007.

[22] A. J. Ricketts, K. Irick, N. Vijaykrishnan, and M. J. Irwin, "Priority scheduling in digital microfluidics-based biochips," in *Proc. Design, Automat., Test Eur.*, 2006, pp. 329–335.

[23] V. Srinivasan, V. K. Pamula, and R. B. Fair, "An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids," *Lab Chip*, vol. 4, pp. 310–315, Aug. 2004.

[24] T. Xu, W. Hwang, F. Su, and K. Chakrabarty, "Automated design of pin-constrained digital microfluidic biochips under droplet-interference constraints," *Assoc. Comput. Machinery J. Emerging Technol. Comput. Syst.*, vol. 3, no. 3, Nov. 2007.

[25] M. G. Pollack, R. B. Fair, and A. D. Shenderov, "Electrowetting-based actuation of liquid droplets for microfluidic applications," *Appl. Phys. Lett.*, vol. 77, no. 11, pp. 1725–1726, 2000.

[26] http://www.liquid-logic.com/technology.html

[27] J. Gong and C. J. Kim, "2-D digital microfluidic system by multilayer printed circuit board," in *Proc. IEEE Micro-Electro-Mechan. Syst.*, 2005, pp. 726–729.

[28] A. Moreno, E. Saridakis, and N. E. Chayen, "Combination of oils and gels for enhancing the growth of protein crystals," *J. Appl. Crystallogr.*, vol. 35, no.1, pp. 140–142, 2002.

[29] W. Hwang, F. Su, and K. Chakrabarty, "Automated design of pin-constrained digital microfluidic arrays for lab-on-a-chip applications," in *Proc. IEEE/Assoc. Comput. Machinery Design Automat. Conf.*, 2006, pp. 925–930.

[30] *Connect-5 Strategies* [Online]. Available: http://www.springfrog.com/games/gomoku/

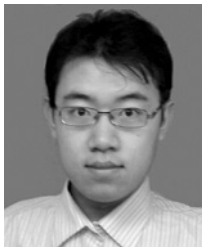
[31] R. Diestel, *Graph Theory*. Berlin, Germany: Springer, 2005.

[32] http://www.pcbdesign.org/pcb-layout/understanding-pcb-layers/

[33] Y. Zhao and K. Chakrabarty, "Cross contamination avoidance for droplet routing in digital microfluidic biochips," in *Proc. IEEE/Assoc. Comput. Machin. Design, Automat., Test Eur. Conf.*, 2009, pp. 1290–1295.

[34] T. Xu and K. Chakrabarty, "Parallel scan-like test and multiple-defect diagnosis for digital microfluidic biochips," *IEEE Trans. Biomed. Circuits Syst.*, vol. 1, no. 2, pp. 148–158, Jun. 2007.

[35] M. G. Pollack, "Electrowetting-based microactuation of droplets for digital microfluidics," Ph.D. dissertation, Dept. Elect. Comput. Eng., Duke Univ., Durham, NC, 2001.



Tao Xu (S'07) received the B.E. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2005, and the M.S. and Ph.D. degrees in electrical and computer engineering from Duke University, Durham, NC, in 2007 and 2008, respectively.

He is currently a Senior Hardware Engineer with Cisco Systems, Inc., Research Triangle Park, NC. He has published 23 papers (6 journal papers and 17 conference papers). His current research interests include design and testing of mixed-technology microsystems, electronic design automation, mixed-signal very-large-scale integration (VLSI) design, micro-electro-mechanical systems modeling and simulation, especially microfluidic biochips.

Dr. Xu is the winner of the Best Paper Award of the 20th International VLSI Conference. He has also given presentations in many conferences including the Design Automation Conference, the Design, Automation and Test in Europe, and the International Test Conference. He is also an invited reviewer for the IEEE TRANSACTION OF COMPUTER AIDED DESIGN OF INTEGRATED CIRCUIT AND SYSTEMS and the *Association for Computing Machinery Journal on Emerging Technology in Computing Systems*.

Krishnendu Chakrabarty (F'08) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 1990, and the M.S.E. and Ph.D. degrees, both in computer science and engineering from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively.

During 1995–1998, he was an Assistant Professor of Electrical and Computer Engineering at Boston University, Boston, MA. He is currently a Professor of Electrical and Computer Engineering with the Department of Electrical and Computer Engineering, Duke University, Durham, NC. He is a Member of the Chair Professor Group (honorary position) in Software Theory with the School of Software, Tsinghua University, Beijing, China. His current research projects include testing and design-for-testability of integrated circuits, digital microfluidics and biochips, circuits and systems based on DNA self-assembly, and wireless sensor networks. He has authored nine books on these topics (including two in press), published over 320 papers in journals and refereed conference proceedings, and given over 130 invited, keynote, and plenary talks.

Dr. Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) Award, the Office of Naval Research Young Investigator Award, the Humboldt Research Fellowship from the Alexander von Humboldt Foundation, Germany, and several best papers awards at IEEE conferences. He is a Golden Core Member of the IEEE Computer Society, and a Distinguished Engineer of Association for Computing Machinery (ACM). He is a 2009 Invitational Fellow of the Japan Society for the Promotion of Science. He is a recipient of the 2008 Duke University Graduate School Dean's Award for excellence in mentoring. He has served as a Distinguished Visitor of the IEEE Computer Society from 2005 to 2007, and as a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2006 to 2007. Currently, he serves as an ACM Distinguished Speaker. He is an Associate Editor of IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON VLSI SYSTEMS, and IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS. He also serves as an Editor of the *Journal of Electronic Testing: Theory and Applications*. He is the Editor-in-Chief of the IEEE DESIGN AND TEST OF COMPUTERS, and the *ACM Journal on Emerging Technologies in Computing Systems*.

Vamsee K. Pamula received the B.E. and M.S. degrees, in 1994 and 1996, respectively. And the Ph.D. degree in electrical and computer engineering from Duke University, Durham, NC, in 2000.

He is a Co-Founder and the Chief Technology Officer of Advanced Liquid Logic, Inc., Research Triangle Park, NC, which develops digital microfluidics for clinical diagnostics, screening, and other research applications. He serves as a principal investigator on several projects funded by the NIH. He has given numerous talks and published dozens of articles on digital microfluidics, authored two book chapters and a book, and has over 70 issued and published patent applications. His current research interests include development of inexpensive and accessible diagnostic devices.