# Automated Design of Pin-Constrained Digital Microfluidic Biochips Under Droplet-Interference Constraints

TAO XU
Duke University
WILLIAM L. HWANG
St John's College, University of Oxford
FEI SU
Intel Corporation
and
KRISHNENDU CHAKRABARTY
Duke University

14

Microfluidics-based biochips, also referred to as lab-on-a-chip, are devices that integrate fluid-handling functions such as sample preparation, analysis, separation, and detection. This emerging technology combines electronics with biology to open new application areas such as point-of-care diagnosis, on-chip DNA analysis, and automated drug discovery. We propose a design automation method for pin-constrained biochips that manipulate nanoliter volumes of discrete droplets on a microfluidic array. In contrast to the direct-addressing scheme that has been studied thus far in the literature, we assign a small number of independent control pins to a large number of electrodes in the biochip, thereby reducing design complexity and product cost. The design procedure relies on a droplet-trace-based array partitioning scheme and an efficient pin assignment technique, referred to as the "Connect-5 algorithm." The proposed method is evaluated using a set of multiplexed bioassays.

This article is an extended and revised version of the paper presented at the 2006 IEEE/ACM Design Automation Conference (DAC) © ACM 2006.
The research work was supported by the National Science Foundation under grants IIS-0312352 and CCF-0541055. Preliminary and abridged versions of this paper were published in Proceedings of the Design, Automation Conference, 2006, pp. 925–930, and in *Proceedings of IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, 2006, pp. 112–117.
William Hwang and Fei Su contributed to this paper while at Duke University.
Author's addresses: T. Xu, Department of Electrical and Computer Engineering, Duke University, Durham, NC; email: tx@ee.duke.edu.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids—*Placement and routing*; B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault Tolerance; J.3 [**Life and Medical Sciences**]: *Biology and genetics; Health*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Physical design automation, droplets, microfluidics, biochips

**ACM Reference Format:**

Xu, T., Chakrabarty, K., Hwang, W. L., and Su, F. 2007. Automated design of pin-constrained digital microfluidic biochips under droplet-interference constraints. ACM J. Emerg. Technol. Comput. Syst. 3, 3, Article 14 (November 2007), 23 pages. DOI = 10.1145/1295231.1295235 http://doi.acm.org/10.1145/1295231.1295235

## 1. INTRODUCTION

Microfluidics-based biochips constitute an emerging technology area that can potentially open up several exciting applications. These devices enable precise control of microliter and nanoliter volumes of biological samples. They combine electronics with biology, and they integrate various bioassay operations, such as sample preparation, analysis, separation, and detection [Verpoorte and Rooij 2003; Schulte et al. 2002; Srinivasan et al. 2004], in a single miniaturized platform. It has been predicted that, by providing miniaturization, automation, and integration, microfluidic biochips will revolutionize laboratory procedures in molecular biology with applications to point-of-care diagnostics, DNA analysis, and automated drug discovery [Schulte et al. 2002; Srinivasan et al. 2004].

Currently, most commercially available biochips either are based on microarrays [Schena 2000] or rely on continuous fluidic flow in etched microchannels [Verpoorte and Rooij 2003]. An alternative design approach utilizes droplets with microliter and nanoliter volumes, thereby obviating the need for cumbersome micropumps and microvalves. Droplets are actuated using on-chip electrodes and moved under the control of a system clock; this microfludic system is similar in operation to a digital microprocessor. Thus, this novel technology is referred to as "digital microfluidics." The "digital" structure also offers reconfigurability and a scalable system architecture based on a two-dimensional array [Pollack et al. 2000; Cho et al. 2002].

A typical biochip consists of a two-dimensional patterned metal electrode array (e.g., chrome or indium tin oxide), on which droplets containing biological samples are dispensed, transported, mixed, incubated, separated, or detected. As bioassays increase in complexity, for example, for high-throughput DNA sequencing [Paegel et al. 2003] and large-scale protein assays for drug discovery [Srinivasan et al. 2004], design tools are needed to map and execute them on the digital micofludic platform. In the next few years, biochip integration and design complexity level are expected to increase significantly. Automated design therefore becomes necessary for this emerging marketplace. An appropriate addressing scheme must be used to activate individual electrodes (unit cells) in the array. Design and CAD research for digital microfluidic biochips has mostly been focused on directly addressable arrays [Chakrabarty and Zeng 2005; Su and Chakrabarty 2004; Su and Chakrabarty 2006; Yuh et al. 2006; Su et al.

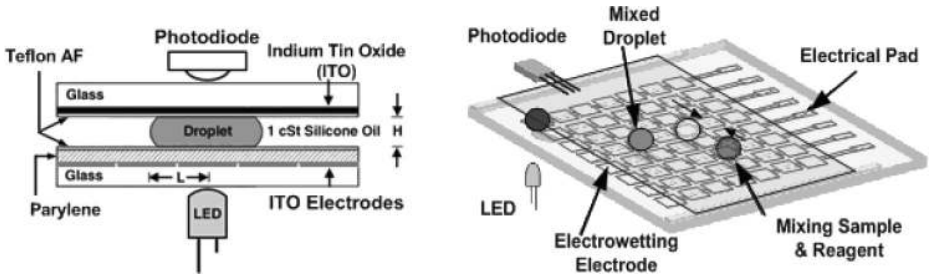



Fig. 1. Schematic diagram of a digital microfluidic biochip.

2006]. In such schemes, each cell of the patterned electrodes can be accessed directly and independently via a dedicated control pin. This method is adequate for small/medium-scale microfluidic electrode arrays (with fewer than $10\times10$ electrodes). However, the number of pins for a design based on direct addressing can be prohibitively high for a large array. For example, a total of $10^4$ pins are needed to independently control the electrodes in a $100\times100$ array. Product cost, however, is a major marketability driver due to the disposable nature of most emerging devices. Hence, simpler wire routing solutions are necessary for electrode addressing.

In this article, we propose an automated digital microfluidic biochip design method based on the partitioning of the microfluidic array and the assignment of a small number of control pins to a large number of electrodes. The partitioning algorithm is based on the concept of "droplet trace," which is extracted from the scheduling and droplet routing results produced by a synthesis tool.

The organization of the rest of the article is as follows. In Section 2, we provide an overview of digital microfluidic biochips. Section 3 discusses related prior work on biochip design automation and pin-constrained system design. Section 4 describes the impact of interference and the electrode addressing problem resulting from control of the array with a reduced number of pins. In Section 5, we propose an array-partitioning-based design method for pin-constrained digital microfluidic biochips. Section 6 evaluates the proposed method using a set of real-life bioassays. Finally, conclusions are drawn in Section 7.

## 2. DIGITAL MICROFLUIDIC BIOCHIPS

A digital microfluidic biochip utilizes the phenomenon of electrowetting to manipulate and move microliter or nanoliter droplets containing biological samples on a two-dimensional electrode array [Pollack et al. 2000]. 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 Figure 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 two-dimensional 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 scheduling results are programmed into a microcontroller that drives electrodes in the array. In addition to electrodes, optical detectors such as LEDs and photodiodes are also integrated in digital microfluidic arrays to monitor colorimetric bioassays [Srinivasan et al. 2004].

To address the need for lowcost, 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 [Gong and Kim 2005]. 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 USB port. However, multiple metal layers for PCB design for large scale microfluidic biochips may lead to reliability problems and increase fabrication cost. Thus, reducing the number of independent control pins is important for successful commercialization. We can also address individual electrodes separately by employing a serial-to-parallel interface. However, this requires active circuit components on the PCB, for example, logic elements such as gates and flip-flops, which will lead to increased cost and power consumption.

## 3. RELATED PRIOR WORK

Recent years have seen growing interest in the design of microfluidic biochips and CAD methods for system design [Su and Chakrabarty 2004; Su and Chakrabarty 2006; Yuh et al. 2006; Su et al. 2006; Gong and Kim 2005; Su et al. 2006]. In Su and Chakrabarty [2004], classical architectural-level synthesis is adapted for automated biochip design based on bioassay protocols. The problem of microfluidic module placement, where array area and fault tolerance serve as the placement criteria, is discussed in Su and Chakrabarty [2006]. A unified synthesis method, which combines operation scheduling, resource binding, and module placement, is proposed in Su and Chakrabarty [2005]. Droplet routing algorithms are presented in Su et al. [2006]. A drawback of these CAD techniques is that they assume a direct-addressing scheme, which requires a large number of independent control pins for large-scale biochips. Thus these methods are unlikely to be useful in practice for low-cost disposable devices.

Pin-constrained design of digital microfluidic biochips was recently proposed and analyzed in Srinivasan et al. [2004]. The number of control pins for a fabricated electrowetting-based biochip is minimized by using a multiphase bus for the fluidic pathways. Every $n$th electrode in an $n$-phase bus is electrically connected. Thus, only $n$ control pins are needed for a transport bus, irrespective of the number of electrodes that it contains. Although the multiphase bus

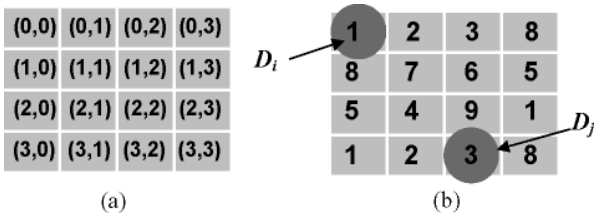



Fig. 2. An example to illustrate droplet interference due to the sharing of control pins by the electrodes: (a) coordinate locations for the electrodes; (b) pin-assignment for the electrodes.

method is useful for reducing the number of control pins, it is applicable only to a one-dimensional (linear) array.

An alternative method based on a cross-reference driving scheme is presented in Fan et al. [2003]. This method allows control of an $N \times M$ grid array with only $N + M$ control pins. The electrode rows are patterned on both the top and bottom plates, and placed orthogonally. In order to drive a droplet along the X-direction, electrode rows on the bottom plate serve as driving electrodes, while electrode rows on the top serve as reference ground electrodes. The roles are reversed for movement along the Y-direction. This cross-reference method facilitates the reduction of control pins. However, it requires a special electrode structure (i.e., both top and bottom plates containing electrode rows), which results in increased manufacturing cost for disposable microfluidic chips. Moreover, this design is not suitable for high-throughput assays because droplet movement is inherently slow.

## 4. IMPACT OF DROPLET INTERFERENCE AND ELECTRODE ADDRESSING PROBLEM

In this section, we formulate the pin-constraint problem for a two-dimensional electrode array. The goal is to use a limited number of independent control pins to control the electrodes of a digital microfluidic array. However, the sharing of control pins leads to the problem of droplet interference, which is defined as the inadvertent activation of multiple electrodes on which droplets are incident at any time instant. Droplet interference results in unintentional droplet operations caused by the simultaneous activation or deactivation of electrodes that are controlled by the same pin.

### 4.1 Impact of Droplet Interference

A pin-constrained layout may result in unintentional droplet movement when multiple droplets are present in the array. Figure 2 shows a $4 \times 4$-array in which the 16 electrodes are controlled by only 9 input pins. The pin numbers are indicated in the figure. Droplet interference occurs if we attempt to move droplet$D_i$ while keeping droplet $D_j$ at its current location. Suppose $D_i$ is at coordinate location (0,0) and $D_j$ is at coordinate location (3,2). To move $D_i$ to (1,0), we need to activate electrode (1,0) and deactivate (0,0). This implies that a high voltage must be applied to Pin 8 while a low voltage must be applied to Pin 1. Note however that a high voltage on Pin 8 also activates electrode (3,3). This results in the inadvertent stretching of droplet $D_j$ across electrodes (3,2) and (3,3).

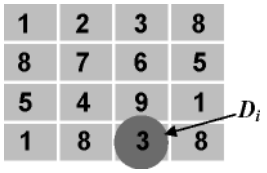



Fig. 3. An example of an inadvertent operation for a single droplet.

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 5 |
| 3 | 5 | 4 | 1 |
| 4 | 1 | 2 | 3 |
| 2 | 3 | 5 | 4 |

Fig. 4. A 5-pin layout for a $4 \times 4$ array.

The sharing of control pins can also affect a single droplet. An example is shown in Figure 3. To move droplet $D_i$ one electrode to the left requires Pin 8 to be activated. However, the electrode on the right of the droplet is also connected to Pin 8; it is therefore also activated. As a result, $D_i$ is pulled from both sides and it undergoes inadvertent splitting. The above example shows that the sharing of control pins can lead to unintentional operations such as droplet splitting and inadvertent movement due to droplet interference. This problem therefore must be avoided in any practical pin-assignment layout.

### 4.2 Minimum Number of Pins for a Single Droplet

Given a two-dimensional microfluidic array, the problem of determining the minimum number of independent control pins, $k$, necessary to have full control of a single droplet without interference can be reduced to the well-known graph-coloring problem [Diestel 2005]. Full control implies that a droplet can be moved to any cell on the array through an appropriate electrode activation sequence. While the problem of finding the chromatic number of a graph is NP-complete [Papadimitriou 1993], it is trivial to observe that for rectangular arrays of size greater than $3 \times 3$, the largest number of directly adjacent neighbors to any cell is four. Hence, if $k$ denotes the number of independent control pins, we ensure that $k \geq 5$ such that each cell and all of its directly adjacent neighbors can be assigned different colors. A possible pin layout using 5 pins for a $4 \times 4$ array is shown as an example in Figure 4.

### 4.3 Pin-Assignment Problem for Two Droplets

We next examine the interference problem for two droplets. For more than two droplets, the interference problem can be reduced to the two-droplet problem by examining all possible pairs of droplets. In general, any sequence of movements for multiple droplets can occur in parallel. We analyze interference between two droplets for a single clock cycle, during which time a droplet can only move to a directly adjacent cell. Any path can be decomposed into unit movements, and

we say that the two paths are compatible if and only if all of their individual steps do not interfere.

In some situations, we would like both droplets to move to another cell in the next clock cycle. If this is not possible without interference, then a contingency plan is to have one droplet undergo a stall cycle (i.e., stay on its current cell). There are other possibilities such as an evasive move or backtracking to avoid interference, but these lead to more substantial changes in the scheduled droplet paths and are therefore not considered in this paper.

Let us denote two droplets by $D_i$ and $D_j$, with the position of droplet $D_i$ at time $t$ given by $P_i(t)$. Let $N_i(t)$ be the set of directly adjacent neighbors of droplet $D_i$. The operator $\bar{k}(\bullet)$ is the set of pins that control the set of cells given by $\bullet$. Then the problem of two droplets moving concurrently can be formally stated as: $D_i$ moves from $P_i(t)$ to $P_i(t+1)$, and $D_j$ moves from $P_j(t)$ to $P_j(t+1)$.

We are interested in the overlap of pins between sets of cells for the interference constraints, rather than the spatial locations of the cells. The latter are important for the fluidic constraints discussed in Su et al. [2006]. For the purpose of defining interference behavior, the system is completely determined by the positional states of the two droplets at times $t$ and $t+1$. For droplet $D_i$,the positional states are characterized by the quartet $(P_i(t), P_i(t+1), N_i(t), N_i(t+1))$ and for $D_j$, the quartet $(P_j(t), P_j(t+1), N_j(t), N_j(t+1))$. We consider $\bar{k}(\bullet)$ of all unordered pairs involving the following sets: $P_i(t)$, $P_i(t+1)$, $N_i(t)$, $N_i(t+1)$, $P_j(t)$, $P_j(t+1)$, $N_j(t)$, $N_j(t+1)$. Since $\binom{8}{2} = 28$, there are 28 such pairs that need to be mutually exclusive to prevent interference between the droplets. Pairs of cell sets that must be mutually exclusive for noninterference must be contained in this pool of unordered pairs because these eight sets include all cells currently occupied by droplets and all of their neighboring cells. A total of 22 pairs can be quickly removed from consideration, leaving only six pairs of sets that need to be closely examined to determine if their mutual exclusion is required for noninterference. For simplicity, the pin operator $\bar{k}(\bullet)$ is left implicit in the following discussion. Mutual exclusion always refers to the *pins* controlling the cells, and not the cells themselves. We can analytically confirm that the control pins for these six pairs must be mutually exclusive to prevent interference.

*Pair 1* $\{P_i(t), N_j(t)\}$: if $N_j(t)$ contains a cell that shares the same pin as $P_i(t)$, then $D_j$ may be between $P_j(t)$ and $P_j(t+1)$ at time $t$. If this is not the case, $D_j$ will not be able to move to $P_j(t+1)$ because it will no longer overlap with $P_j(t+1)$.

*Pair 2* $\{P_i(t+1), N_j(t)\}$: if $N_j(t)$ contains a cell that shares the same pin as $P_i(t+1)$, $D_j$ will not move properly at time $t+1$ unless $P_j(t+1)$ is the cell in $N_j(t)$ that shares the same pin as $P_i(t+1)$.

*Pair 3* $\{P_i(t+1), N_j(t+1)\}$: if $N_j(t+1)$ contains a cell that shares the same pin as $P_i(t+1)$, then $D_j$ will drift after moving to $P_j(t+1)$ so that it ends up between $P_j(t+1)$ and a cell that is an element of $N_j(t+1)$.

*Pair 4* $\{N_i(t), P_j(t)\}$: if $N_i(t)$ contains a cell that shares the same pin as $P_j(t)$, then at time $t$, $D_i$ will drift between $P_i(t)$ and the cell of interest in $N_i(t)$.

$D_i$ may no longer overlap with $P_j(t+1)$; if so, it will not be able to move to $P_j(t+1)$.

*Pair 5* $\{N_i(t),\ P_j(t+1)\}$: if $N_i(t)$ contains a cell that shares the same pin as $P_j(t+1)$, $D_i$ will not move properly at time $t+1$ unless $P_i(t+1)$ is the cell in $N_i(t)$that shares the same pin as $P_j(t+1)$.

*Pair 6* $\{N_i(t+1),\ P_j(t+1)\}$: if $N_i(t+1)$ contains a cell that shares the same pin as $P_j(t+1)$ then at time $t+1$, $D_i$ will drift between $P_i(t+1)$ and the cell of interest in $N_i(t+1)$.

It can be easily shown that the control pins for these six pairs must be mutually exclusive to prevent interference between droplets $D_i$ and $D_j$. These lead to the following necessary and sufficient interference constraints:

$$
\begin{aligned}
&(1)\ \bar{k}\left(\{P_i(t)\}\right) \cap \bar{k}\left(N_j(t)\right) = \phi\\
&(2)\ \bar{k}\left(\{P_i(t+1)\}\right) \cap \bar{k}\left(N_j(t+1)\right) = \phi\\
&(3)\ \bar{k}\left(N_i(t)\right) \cap \bar{k}\left(\{P_j(t)\}\right) = \phi\\
&(4)\ \bar{k}\left(N_i(t+1)\right) \cap \bar{k}\left(\{P_j(t+1)\}\right) = \phi\\
&(5)\ \bar{k}\left(\{P_i(t+1)\}\right) \cap \bar{k}\left(N_j(t) - \{P_j(t+1)\}\right) = \phi\\
&(6)\ \bar{k}\left(\{P_j(t+1)\}\right) \cap \bar{k}\left(N_i(t) - \{P_i(t+1)\}\right) = \phi.
\end{aligned}
$$

Moving one droplet and stalling the other is a special case that can be used when concurrent movement of two droplets leads to violation of the interference constraints. In this situation, $P_j = P_j(t) = P_j(t+1)$ and the interference constraints reduce to:

$$
\begin{aligned}
&(1)\ \bar{k}\left(\{P_i(t)\}\right) \cap \bar{k}\left(N_j\right) = \phi\\
&(2)\ \bar{k}\left(\{P_i(t+1)\}\right) \cap \bar{k}\left(N_j\right) = \phi\\
&(3)\ \bar{k}\left(N_i(t)\right) \cap \bar{k}\left(\{P_j\}\right) = \phi\\
&(4)\ \bar{k}\left(N_i(t+1)\right) \cap \bar{k}\left(\{P_j\}\right) = \phi\\
&(5)\ \bar{k}\left(\{P_i(t+1)\}\right) \cap \bar{k}\left(N_j - \{P_j\}\right) = \phi\\
&(6)\ \bar{k}\left(\{P_j(t+1)\}\right) \cap \bar{k}\left(N_i - \{P_i\}\right) = \phi.
\end{aligned}
$$

The given constraints must be satisfied during droplet routing [Su et al. 2006; Xu and Chakrabarty 2007]. Therefore, they are enumerated here for the sake of completeness.

## 5. ARRAY PARTITIONING AND PIN-ASSIGNMENT METHODS

In this section, we propose a pin-constrained design method for digital microfluidic biochips based on array partitioning. The key idea is to "virtually" partition the array into regions. Similar partitioning techniques have been used for VLSI circuits and for microarray biochips [Breuer 1977; Dunlop and Kernighan 1985; Kahng et al. 2003]. At any given time, partitions use nonoverlapping sets of pins. Array partitioning may be applied either before or after synthesis. If the biochip is intended for a specific application, or a known set of bioassays, array partitioning and pin assignment can be carried out post-synthesis, that is, after the bioassays are mapped to the digital microfluidic platform. On the other hand, if the target applications are not known in advance, array partitioning must be done prior to synthesis.

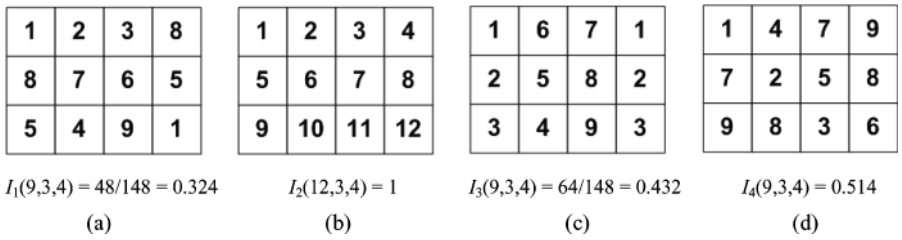



Fig. 5. (a) 9-pin layout, (b) reference pin layout, (c) alternative 9-pin layout, and (d) second alternative 9-pin layout for a $3 \times 4$ array.

### 5.1 Pre-Synthesis Array Partitioning

In some scenarios, chip designers will either have little information about target applications, or the target applications will be diverse. In such scenarios, it is desirable to have an application-independent pin-constrained layout with a low probability of droplet interference. This goal can be achieved using pre-synthesis array partitioning.

Given $k$ independent pins and no synthesis data (scheduling of operations, placement of modules, droplet pathways, etc.), pre-synthesis array partitioning aims at maximizing the number of independent movements that a droplet can undertake from each position of the array while not interfering with another droplet on the same array. If interference results, some droplets can be forced to wait (stall cycle) while others move. If interference still persists, then there is no safe way to transport the droplets on the same array along their desired paths unless the schedule is modified. In order to develop a useful, application-independent index that represents the independence of movement for two droplets on an array that can be easily extended to multiple droplets, we need a metric for comparison. If every cell has a dedicated control pin, there is no possibility of interference between droplets. Hence, only the fluidic constraints need to be considered.

We consider the case of one droplet moving while the other droplet waits. Let $\Phi$ be the set of all possible pin configurations using $k$ pins for an $n \times m$ array. For a particular pin configuration $c \in \Phi$ using $k$-pins in our 2-droplet system, let $I_c(k,n,m)$ denotes the pin-constrained noninterference index (PCNI). It takes a value between 0 and 1 and equals the fraction of legal moves for two droplets (one moving, one waiting) on an $n \times m$ array with each cell having its own dedicated control pin that are still legal with pin layout $c \in \Phi$ and $k < n \times m$ pins, that is,

$$I_c(k,n,m) = \frac{number\ of\ legal\ moves\ in\ configuration\ c\ with\ k < n \times m\ pins}{number\ of\ legal\ moves\ with\ k = n \times m\ pins}.$$

The interference and fluidic constraints discussed above can be used to determine $I_c(k,n,m)$ for a given pin layout. The computational complexity of this procedure is $O(n^2m^2)$. For example, consider the $3 \times 4$ array with the 9-pin layout shown in Figure 5(a). Using the interference and fluidic constraints, we determine that there are a total of 48 legal moves for the given array dimensions and pin layout.

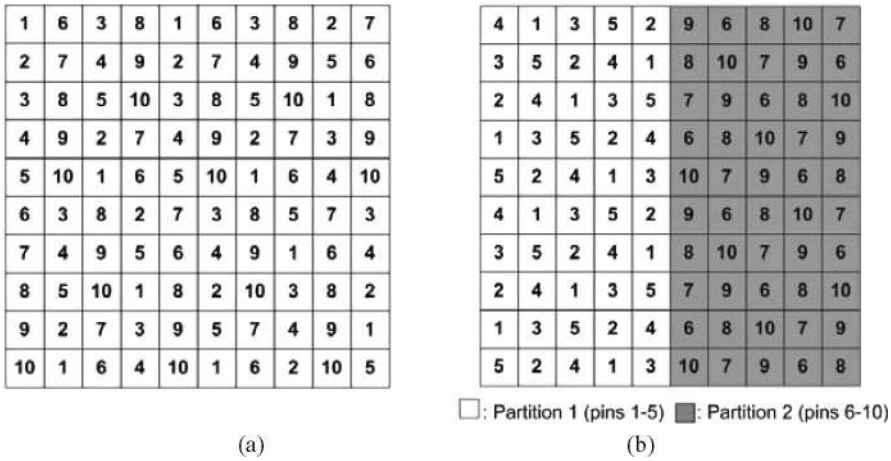



Fig. 6. (a) Nonpartitioned and (b) two-partition 10-pin layout for a 10 × 10 array.

The reference pin layout, which provides full controllability with 12 independent pins and yields 148 legal moves, is shown in Figure 5(b). Therefore, the fraction of legal moves for the reference pin layout that are still legal for the proposed pin layout is 48/148 = 0.324. Two other layouts and their PCNIs are illustrated in Figures 5(c) and 5(d). Better layouts seem to obey two principles: 1) Spread out placement of pins that are used multiple times, and 2) Place pins that are more often used on cells that have fewer neighbors (e.g., sides and corners).

For the case of two droplets moving concurrently, we use the same definition as before of PCNI, except that a legal move is now characterized by a unique quartet, $(P_i(t), P_i(t+1), P_j(t), P_j(t+1))$ that satisfies the fluidic and interference constraints. For example, for the same 3×4 array with the pin layout shown in Figure 5(a), there are a total of 88 legal moves. The reference pin layout, shown in Figure 5(b), yields 264 legal moves. Therefore, the fraction of legal moves for the reference pin layout that are still legal for the proposed pin layout is 88/264 = 0.333.

Next we discuss the impact of array partitioning on PCNI. We first start from a specific example and then expand to general cases.

Consider the nonpartitioned 10 × 10 array with $k = 10$ shown in Figure 6(a). The noninterference index is 0.1467. If we instead divide the array into two partitions of 5 pins each, still with $k = 10$, then one possible layout is shown in Figure 6(b). The noninterference index increases to 0.4565 for the 2-partition array, an increase of 211.18% relative to the nonpartitioned array. In general, as the array size increases, and the number of pins and number of partitions remain constant, the PCNI for a partitioned array will increase while that of a nonpartitioned array will decrease. The reasons for this are the following: (1) for a nonpartitioned array, the possibilities for interference increase as array size increases and pins number remains the same, and (2) for a partitioned array, the partition size increases with the array size if $k$ and the number of partitions are constant. Larger partitions mean that the proportion of movement pairs that

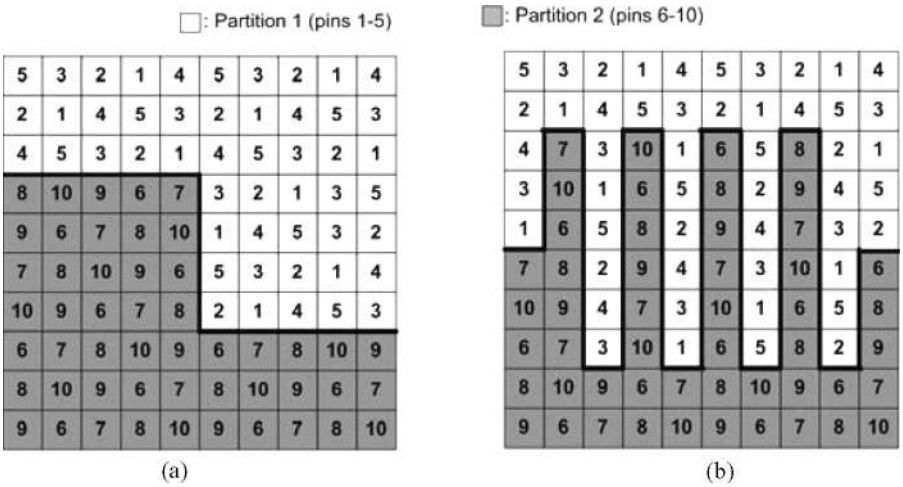


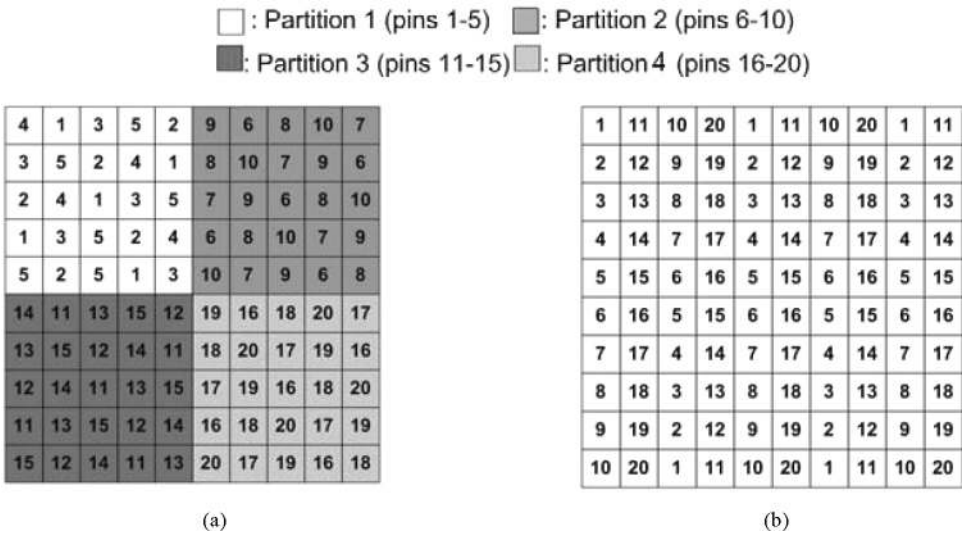

Fig. 7. Two-partition layouts for a 10 × 10 array with (a) medium boundary length, and (b) long boundary length.

violate fluidic constraints at the borders decreases, while interference is more rarely a concern for a partitioned array.

Figure 7(a) shows the same $10 \times 10$ array with the two partitions rearranged. The PCNI value is 0.4215, which is slightly less than the value of 0.4565 obtained with the vertical-boundary partition configuration in Figure 6(b).

It is interesting to examine why the second partitioning scheme results in a lower PCNI. The boundary length between the partitions dictates the number of opportunities for fluidic interference (i.e., inadvertent mixing) between two droplets even when they satisfy the "rule" of at most one droplet per partition at any given time. Compared to the partitioning scheme of Figure 6(b), which has a boundary length of 10 edges, the partitioning scheme of Figure 7(a) introduces four more edge electrodes, on which inadvertent mixing can occur. Some droplet pathways that route the droplets to these four edges may be forbidden due to violation of fluidic constraints, thereby resulting in a reduced number of legal moves. To illustrate this further, the example in Figure 7(b) also has two partitions for a $10 \times 10$ array with $k = 10$. The boundary length is 58. As expected, the PCNI drops significantly to 0.2478 with the large increase in boundary length, but it still represents a 68.92% increase relative to the nonpartitioned array. These observations lead us to conclude that, to maximize the PCNI, the boundary length between partitions should be kept to the minimum possible for a given assay.

To illustrate the use of more than two partitions, consider the array pin layout with $k = 20$ and 4 partitions, as shown in Figure 8(a). The two-droplet PCNI is 0.7331 relative to the nonpartitioned array shown in Figure 8(b), which yields an index of 0.6772. It is important to note that since the PCNI reported here is only computed for two droplets simultaneously on the array, the advantages of the 4-partition layout in Figure 8(a) are not being fully exploited. Whereas four droplets maintained in unique partitions will have no potential for interference




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 1 | 3 | 5 | 2 | 9 | 6 | 8 | 10 | 7 |
| 3 | 5 | 2 | 4 | 1 | 8 | 10 | 7 | 9 | 6 |
| 2 | 4 | 1 | 3 | 5 | 7 | 9 | 6 | 8 | 10 |
| 1 | 3 | 5 | 2 | 4 | 6 | 8 | 10 | 7 | 9 |
| 5 | 2 | 5 | 1 | 3 | 10 | 7 | 9 | 6 | 8 |
| 14 | 11 | 13 | 15 | 12 | 19 | 16 | 18 | 20 | 17 |
| 13 | 15 | 12 | 14 | 11 | 18 | 20 | 17 | 19 | 16 |
| 12 | 14 | 11 | 13 | 15 | 17 | 19 | 16 | 18 | 20 |
| 11 | 13 | 15 | 12 | 14 | 16 | 18 | 20 | 17 | 19 |
| 15 | 12 | 14 | 11 | 13 | 20 | 17 | 19 | 16 | 18 |

(a)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 11 | 10 | 20 | 1 | 11 | 10 | 20 | 1 | 11 |
| 2 | 12 | 9 | 19 | 2 | 12 | 9 | 19 | 2 | 12 |
| 3 | 13 | 8 | 18 | 3 | 13 | 8 | 18 | 3 | 13 |
| 4 | 14 | 7 | 17 | 4 | 14 | 7 | 17 | 4 | 14 |
| 5 | 15 | 6 | 16 | 5 | 15 | 6 | 16 | 5 | 15 |
| 6 | 16 | 5 | 15 | 6 | 16 | 5 | 15 | 6 | 16 |
| 7 | 17 | 4 | 14 | 7 | 17 | 4 | 14 | 7 | 17 |
| 8 | 18 | 3 | 13 | 8 | 18 | 3 | 13 | 8 | 18 |
| 9 | 19 | 2 | 12 | 9 | 19 | 2 | 12 | 9 | 19 |
| 10 | 20 | 1 | 11 | 10 | 20 | 1 | 11 | 10 | 20 |

(b)

Fig. 8. Partitioned pin layout for a $10 \times 10$ array with $k = 20$ and 4 partitions; (b) nonpartitioned pin layout for a $10 \times 10$ array.

and therefore the four-droplet PCNI will only drop slightly for the partitioned array relative to the two-droplet PCNI, the four-droplet PCNI will drop precipitously for the nonpartitioned array. It is remarkable that even if we ignore this obvious advantage, the PCNI of the partitioned array is still 8.25% higher than that of the nonpartitioned array. Even though the virtual partitions are not considered in the computation of the PCNI, the array partitioning scheme still leads to significantly higher indices relative to nonpartitioned arrays. Therefore, it is generally advantageous to partition the array even if the assays of interest use the array in a random fashion, that is, with no regard to the partitions.

The proposed pre-synthesis array partitioning method is application-independent. Therefore, the partitioned array with a high PCNI value can be used for a large variety of bioassays. Few changes need to be made to the synthesis results for these assays and the impact on execution time is therefore negligible.

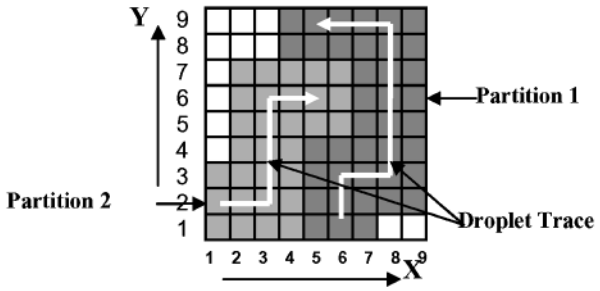
### 5.2 Post-Synthesis Partitioning

In a second scenario, biochip designers may focus on a specific bioassay. In this scenario, the goal is to address electrodes in the synthesized microfluidic array with no droplet interferences and with the minimum number of control pins, thereby reducing the product cost. In this subsection, we propose a post-synthesis array partitioning method that meets all of these requirements.

Recall that mutually exclusive sets of pins are utilized for different partitions. Therefore, if we can partition the array so that droplets are in different partitions, interference between them can be avoided. Partitions can be viewed as subarrays that can contain at most one droplet at any given time. Hence, the

| | Detector1(x,y) | Detector2(x,y) | Detector3(x,y) |
|---|---|---|---|
| Droplet 1 | (8, 3) | (8, 9) | (5, 9) |
| Droplet 2 | (3, 2) | (3, 6) | (5, 6) |

(a)




(b)

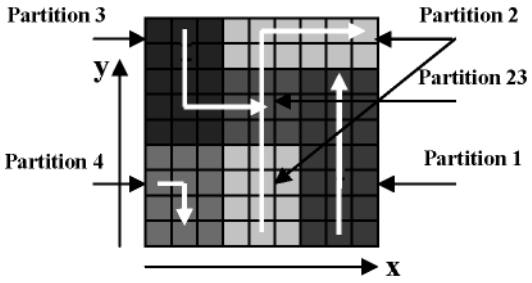
Fig. 9. (a) Detectors used in bioassay; (b) Routing result and array partitions.

partitioning criterion here is to ensure at most one droplet is included in each partition. However, partitions with no droplets (at any point in time) should be avoided because no droplet manipulation is done in this region with the additional set of pins assigned to it. Hence it is best to ensure that each partition has exactly one droplet in it.

Based on this requirement, we find that the droplet trace, defined as the set of cells traversed by a single droplet, serves as a good tool for generating the array partitions. Since we view pin assignment as the last step in system synthesis, information about module placement and droplet routing is available a priori. The droplet trace can be easily extracted from the droplet routing information and the placement of the modules to which it is routed. A trace extraction example is shown in Figure 9, where two droplets are to be manipulated on the microfluidic array. Both of these are required to be detected by an optical sensor three times in a specific bioassay. The placement of these detectors is shown in Figure 9(a). The droplet routes, that is, the path taken by droplets, are shown by the arrows in Figure 9(b). The connected arrows illustrate the traces of the two droplets. For each droplet, we create a partition composed of all the cells on its trace as well as the cells adjacent to the trace. The adjacent cells are included to form a "guard ring" along the trace to avoid inadvertent mixing and movement. The guard rings are a consequence of the fluidic constraint described in Su et al. [2006].

Note that in Figure 9(b), there are two "white" regions that belong to neither partition. They are referred to as "don't-care" regions because they are similar to the "don't-care" terms in logic synthesis; either they can be assigned to any partition or they can together form an additional partition if multi-droplet-operation modules, for example, mixers, can be positioned in them.

In order to reduce the number of partitions, we introduce a time-division pin-sharing method. The basic idea is to merge partitions that have no overlapping




(a)

| Partition | Time Span |
|---|---|
| 1 | 1-7 |
| 2 | 5-16 |
| 3 | 7-14 |
| 4 | 17-20 |
| 23.2 | 10-11 |
| 23.3 | 13-14 |

(b)

Fig. 10. (a) Routing result and partitioning (b) Time-span table for the droplets.

time spans, where a *time span* for a partition is defined as the period of time during which it contains a droplet. The time spans for all the partitions can be easily calculated from the operation schedule, module placement and droplet routing results [Su et al. 2006]; the overlaps can then be readily determined. Partitions with nonoverlapping time spans are merged to form a larger partition. This check-merge procedure continues until all partition pairs overlap in their time spans. By reducing the number of partitions, we can reduce the number of control pins needed for the array. Note that droplet traces may have spatial overlap, that is, they may intersect at one or more unit cells on the array. In this case, the requirement of one droplet per partition is not met and droplet interference may occur. This problem is handled by simply modifying the partitioning result.

We next study the case where droplets traces intersect on the array. This implies that partitions derived by the proposed method overlap in some regions. Sets of pins from an "overlapping" partition cannot be used in the overlapped region since the reuse of the pins may lead to droplet interference. One solution to this problem is to make the overlapping region a new partition, referred to as the overlapping partition, and use direct-addressing for it. Again, time-division pin-sharing (TDPS) can be used to reduce the number of pins since pin sets of the other (nonoverlapping) partitions can be candidates for direct-addressing in the overlapping partition.

An example of this approach is shown in Figure 10. The droplet traces are first derived from the droplet routing information. Partitions 1, 2, 3, and 4 are assigned accordingly. Partition 2 and Partition 3 overlap with each other as shown. Thus a new Partition 23 is created. From the scheduling result in Figure 10(b), the time span for Partition 23 is found to be 10-14s. Next the time spans for Partitions 1 and 4 are checked and it is seen that their time spans do not overlap with that for Partition 23. Hence the two sets of pins (a total of $2 \times 5 = 10$ pins) in Partitions 1 and 4 can be used to directly address the nine electrodes in Partition 23.

Partitions that share pins with the overlapping partition are empty while droplets are manipulated in the overlapping partition. Therefore, the sharing of pins in these cases does not lead to droplet interference. By introducing the concept of TDPS, we can significantly reduce the number of pins required for

Table I. Time-Span Table with Detailed Scheduling Results for the Overlapping Region

| Partition | Time Span |
|---|---|
| 1 | 1–7 |
| 2 | 5–12 |
| 3 | 7–23 |
| 4 | 17–20 |
| 23 | 10–14 |

independent addressing in overlap partitions. The concept of TDPS can also be applied in the spatial dimension to the operations inside the overlapping region to further reduce the number of control pins.

Once a spatially overlapping region is found, we determine if there are temporally overlapping droplets in this region. Depending on the outcome of this procedure, a spatial overlap region can be then divided into two groups—a spatially overlapping but temporally nonoverlapping (SOTN) region, and a spatially overlapping as well as temporal overlapping (SOTO) region. For SOTO regions, direct addressing is used. For SOTN regions, even though droplets traces cross each other, different droplets are sequenced in time (one after the other), that is, at any point in time, there is at most one droplet inside the region. In this case, a pin set with the minimum size ($k = 5$) for single droplet manipulation is assigned to this SOTN region.

Again, we use the above example of Figure 10 for illustration. Table I shows the schedule information needed for carrying out the temporal check for the overlapping region. Partitions 23.2 and 23.3 represent the manipulation of Droplet 2 and Droplet 3 in Partition 23 respectively. Table I shows that the time spans for these partitions do not overlap, thus five pins (in contrast to the nine pins needed for direct-addressing) are adequate for the overlapping partition.

### 5.3 Pin-Assignment Algorithm

In Section 5.1 and Section 5.2, we have discussed two partitioning methods (pre- and post-synthesis) for digital microfluidic arrays. In both methods, each partition is assigned a distinct pin set. In this section, we address the problem of how to map control pins to the electrodes in a partition. An efficient algorithm that can be easily implemented using a 3-layer-PCB is presented. The algorithm is based on a strategy of the Connect-5 (Gomoku) board game [Connect5 strategies], thus it is referred to as the Connect-5 algorithm.

The sets of pins assigned to the partitions belong to two groups according to their cardinality, that is, the minimum for single droplet manipulation ($k = 5$) or the number of pins required for direct-addressing. Here we focus on the pin assignment problem for the first case, since pin assignment for direct-addressing is straightforward (there exists a simple one-to-one mapping between pins and electrodes).

Our goal is to ensure that any five adjacent unit cells (i.e., a central cell and its four neighbors) that form a "cross" are assigned distinct pins. We refer to the

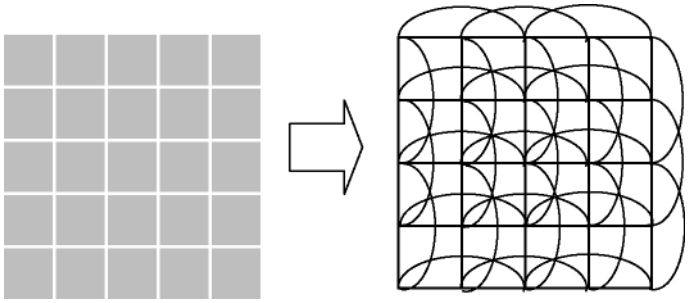


Fig. 11. Mapping of an array to an undirected graph.

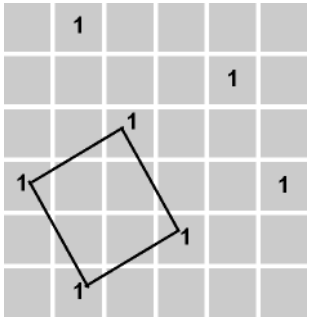



Fig. 12. A single Bagua structure (the tilted square) and its repetition in a square partition.

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 [Diestel 2005]. The problem here is to obtain a 5-coloring of the graph derived from a partition, as shown in Figure 11. The unit cells in the partition are mapped to vertices and any two cells that belong to a "cross" are connected by an edge. The graph corresponding to a partition is referred to as the partition graph.

The graph coloring problem, which involves the determination of the chromatic number $\chi(G)$ for a graph $G$, is known to be NP-complete [Papadimitriou 1993]. However, if $\chi(G)$ or the number of colors to be used is known, as is the case here, there exists an efficient algorithm for graph coloring. Moreover, the regular structure of the partitions 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 [Connect 5 strategies]. A Bagua is a tilted square, as shown in Figure 12. By repeatly placing Bagua structures next to each other until the partition boundaries are reached, a Bagua repetition is derived as shown in Figure 12. The tiling using Bagua repetitions forms the basis for the Connect-5 algorithm.

Five copies of Bagua repetitions are sufficient to cover a partition of any size. This is because of the following property of a Bagua repetition: vertices

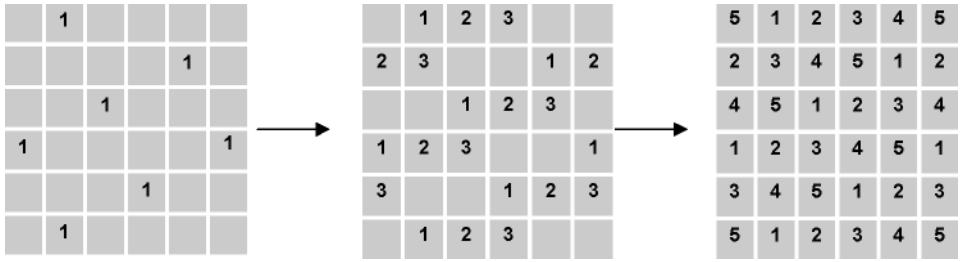



Fig. 13. Covering a partition by shifting Bagua repetition along rows.

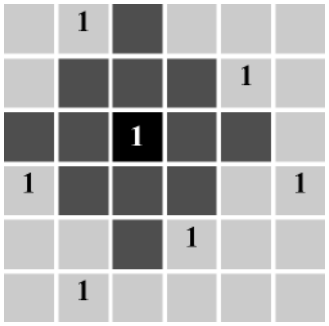



Fig. 14. A demonstration that the "cross constraints" are met.

connected to the same (shared) pin appear after exactly five cells in the same row or column of the partition. The partition can be covered with Bagua repetitions by simply taking a Bagua repetition and shifting it one cell along an arbitrary direction, for example, upwards, then assigning it to another control pin and repeating this step four times, as shown in Figure 13. Note that, although the shifting direction is arbitrarily selected at the start of the tiling process, once chosen it must be consistent over the four shifting steps.

As shown in Figure 13, the pin assignment that results from the shifting of Bagua repetition satisfies a cyclic property, that is, each row is a cyclic repetition of an ordered sequence, and it is also a shifted copy (shift by two cells) of the previous row. This cyclic property provides an easy way to implement the Connect-5 algorithm.

To start, the first row of a partition is selected. Pins are assigned in a fixed cyclic order until the boundary of the partition is reached. Then in the next row, the same order is used but with a 2-cell-shift to the left/right. The procedure continues until all cells in the partition have been assigned pins. Recall that the shifting direction, once chosen, must remain fixed during the assignment procedure for a given partition.

Next we show that control pins assigned to the electrodes in a partition using this method allow free movement of a single droplet; that is, the "cross constraint" is met. To demonstrate this, we consider the cell which is hatched in Figure 14. 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

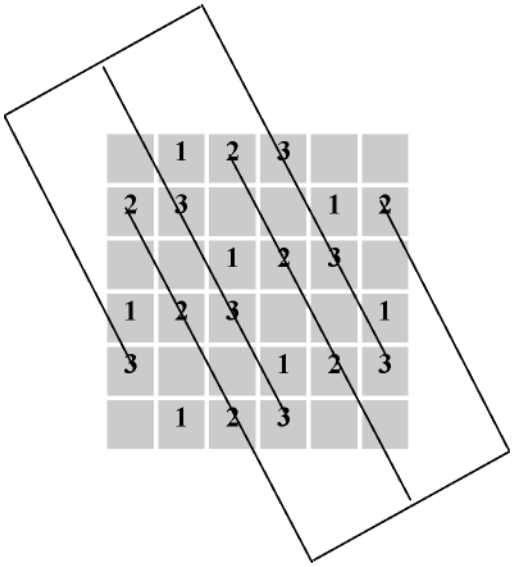



Fig. 15. A wiring example for the pin assignment obtained using the Connect-5 algorithm. For each partition, two pins can be wired in one layer.

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 partition can be tiled by five copies of Bagua repetitions, the "cross constraint" is automatically met for every cell in our pin assignment method.

Compared to the graph coloring approach, the Connect-5 algorithm offers the important advantage that it allows wiring to be done easily on a 3-layer PCB; see Figure 15. The graph coloring approach does not lend itself to this simple pin layout because of the likelihood of irregular vertex coloring.

Connect-5 algorithm succeeds in avoiding droplet interference while moving a single droplet inside the partition. Recall that in the post-synthesis array partitioning method, partitions contains only one droplet each. Therefore, the Connect-5 algorithm can be integrated into the post-synthesis array partitioning method to generate droplet-interference-free layouts with a minimum number of pins. For pre-synthesis array partitioning, where multiple droplets might be present within one partition, the Connect-5 algorithm can reduce the probability of interference by eliminating the interference for every single droplet.

## 6. APPLICATION TO MULTIPLEXED BIOASSAY

To evaluate the array partitioning and pin assignment method for pin-constrained microfluidic biochips, we use a real-life experiment of a multiplexed biochemical assay consisting of a glucose assay and a lactate assay based on colorimetric enzymatic reactions. These assays have been demonstrated recently [Srinivasan et al. 2004]. The digital microfluidic biochip contains a $15\times15$ microfluidic array, as shown in Figure 16. The schedule for the set of bioassays, if

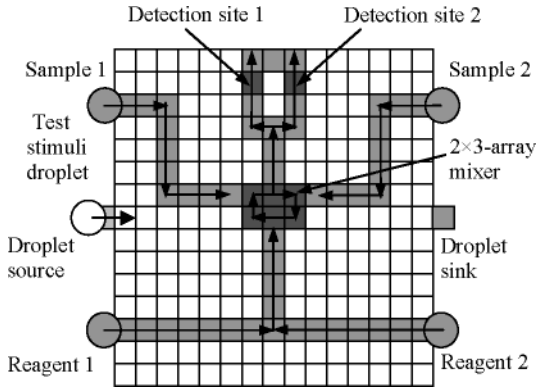



Fig. 16. A 15 × 15 array used for multiplexed bioassays.

Table II. Bioassay Schedule for a Multiplexed Bioassay

| Step/Time Elapsed (s) | Operation |
|---|---|
| Step 1 / 0 | Sample 2 and Reagent 2 start to move towards the mixer. |
| Step 2 / 0.8 | Sample 2 and Reagent 2 begin to mix together and turn around in the 2 × 3-array mixer. |
| Step 3 / 6.0 | Sample 1 and Reagent 1 start to move towards the mixer. Sample 2 and Reagent 2 continue the mixing. |
| Step 4 / 6.8 | Sample 2 and Reagent 2 finish the mixing and product 2 leaves the mixer to optical detection location 2.<br>Sample 1 and Reagent 1 begin to mix in the 2 × 3-array mixer. |
| Step 5 / 12.8 | Sample 1 and Reagent 1 finish the mixing and product 1 leaves the mixer to the optical detection location 1.<br>Product 2 continues the absorbance detection. |
| Step 6 / 19.8 | Product 2 finishes optical detection and leaves the array to the waste reservoir.<br>Product 1 continues the absorbance detection. |
| Step 7 / 25.8 | Product 1 finishes optical detection and leaves the array to the waste reservoir. One procedure of the multiplexed bioassays ends. |

a microfludic array with 225 control pins is available, is listed in Table II; one iteration of the multiplexed assays takes 25.8 seconds [Srinivasan et al. 2004]. The movement of droplets is controlled using a 50 V actuation voltage with a switching frequency of 16 Hz. A depiction of the droplet paths for multiplexed glucose and lactase assays is shown in Figure 16.

### 6.1 Results for Pre-Synthesis Array Partitioning

We first apply the pre-synthesis array partitioning method to the array. As discussed in Section 5.1, the corresponding pin layout is independent of the synthesis results. The array partition in this case is not derived from the droplet traces for the synthesized biochip. We use the same array partitions as in Figure 8(a). The 15×15 array is divided into four partitions with 5 pins assigned in each partition using Connect-5 algorithm, as shown in Figure 17.

The partitioned array in Figure 17 results in a PCNI value of 0.7547. Next we use the pin-assignment layout in Figure 17 to execute the synthesized multiplex

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 1 | 2 | 11 | 12 | 13 | 14 | 15 | 11 | 12 | 13 |
| 3 | 4 | 5 | 1 | 2 | 3 | 4 | 13 | 14 | 5 | 11 | 12 | 13 | 14 | 15 |
| 5 | 1 | 2 | 3 | 4 | 5 | 1 | 15 | 11 | 12 | 13 | 14 | 15 | 11 | 12 |
| 2 | 3 | 4 | 5 | 1 | 2 | 3 | 12 | 13 | 14 | 15 | 11 | 12 | 13 | 14 |
| 4 | 5 | 1 | 2 | 3 | 4 | 5 | 14 | 15 | 11 | 12 | 13 | 14 | 15 | 11 |
| 1 | 2 | 3 | 4 | 5 | 1 | 2 | 11 | 12 | 13 | 14 | 15 | 11 | 12 | 13 |
| 3 | 4 | 5 | 1 | 2 | 3 | 4 | 13 | 14 | 15 | 11 | 12 | 13 | 14 | 15 |
| 5 | 1 | 2 | 3 | 4 | 5 | 1 | 15 | 11 | 12 | 13 | 14 | 15 | 11 | 12 |
| 8 | 9 | 10 | 6 | 7 | 8 | 9 | 18 | 19 | 20 | 16 | 17 | 18 | 19 | 20 |
| 10 | 6 | 7 | 8 | 9 | 10 | 6 | 20 | 16 | 17 | 18 | 19 | 20 | 16 | 17 |
| 7 | 8 | 9 | 10 | 6 | 7 | 8 | 17 | 18 | 19 | 20 | 16 | 17 | 18 | 19 |
| 9 | 10 | 6 | 7 | 8 | 9 | 10 | 19 | 20 | 16 | 17 | 18 | 19 | 20 | 16 |
| 6 | 7 | 8 | 9 | 10 | 6 | 7 | 16 | 17 | 18 | 19 | 20 | 16 | 17 | 18 |
| 8 | 9 | 10 | 6 | 7 | 8 | 9 | 18 | 19 | 20 | 16 | 17 | 18 | 19 | 20 |
| 10 | 6 | 7 | 8 | 9 | 10 | 6 | 20 | 16 | 17 | 18 | 19 | 20 | 16 | 17 |

Fig. 17. Pin layout obtained from pre-synthesis array partitioning and pin-assignment.

bioassay. A few changes must be made to the design obtained from the synthesis procedure. For example, in pre-synthesis partitioning, Detection Site 1 and the routing path for Sample 1 (from reservoir to the mixer) are allocated to the same partition. Therefore, the routing of a droplet of Sample 1 to the mixer and the detection of a mixed droplet at Site 1 cannot be carried out simultaneously because of the potential droplet interference. To solve this problem, several stall cycles are added to the schedule and the completion time is extended to 36.8 s, an increase of 50%. However, the increase in assay completion time is accompanied by a significant reduction in the number of pins required (from $15 \times 15 = 225$ pins to $4 \times 5 = 20$ pins), and an application-independent pin layout.

### 6.2 Results for Post-Synthesis Array Partitioning

We next apply the post-synthesis array partitioning method to the multiplexed bioassay example. Initially, six partitions are created for the four droplet traces of Reagents 1, 2 and Samples 1, 2, and the two traces corresponding to the mixed samples going to Detector 1 and Detector 2. Three more partitions are created for the three trace-overlapping regions respectively. Next, time-span overlap is checked for the three spatially overlapping partitions (Partitions 3, 4 and 5). Since there is no temporal overlap of droplets being manipulated in both Partition 3 and Partition 5, only five pins are needed for each of them. Partition 4 is recognized as a mixer, thus only five pins are needed for it. In the next step, time span overlap is checked for all partition pairs. The six partitions corresponding to four droplets traces and two detector paths merge into two partitions (Partition 1 and Partition 2). Finally, the Connect-5

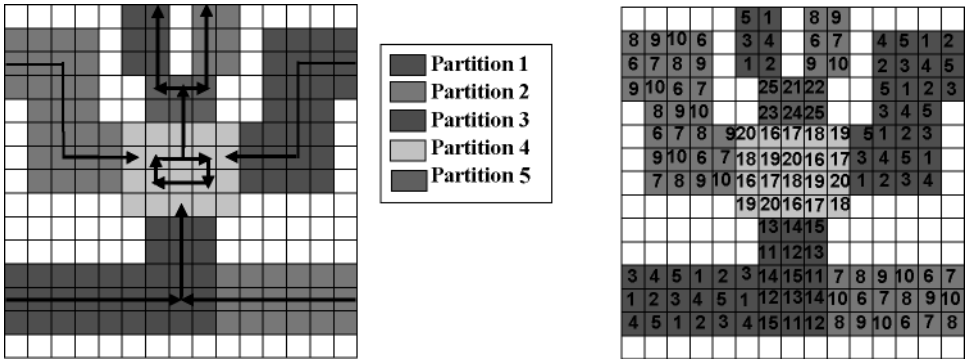



Fig. 18. Partition and pin assignment results for the multiplexed bioassay. Blank areas are don't-care regions that can be either left unaddressed or combined with any other partition.

algorithm is applied. The partitions and the pin-assignment results are shown in Figure 18.

We therefore see that array partitioning and pin assignment are effective in reducing the input bandwidth, while maintaining the same throughput that is obtained for a directly addressable array. Five partitions are sufficient for preventing interference between multiple droplets on the array, as shown in Figure 18. Since only five control pins are necessary for full control of a single droplet within each partition, only 25 out of the possible 225 control pins are necessary, i.e., only 11.11% of the total number of electrodes. This represents a significant reduction in input bandwidth without sacrificing throughput.

To compare with the pre-synthesis partitioning and pin-assignment method, we calculate the PCNI value for the layout in Figure 18. Each "unassigned" electrode is randomly connected to one of the 25 pins. We obtain a PCNI value of 0.5743, while the pre-synthesis array partitioning achieves a PCNI value of 0.7547. We therefore see that although post-synthesis array partitioning generates a more efficient layout for a specific bioassay, pre-synthesis array partitioning is more suitable when the microfluidic array needs to be used for several target applications that are not known *a priori*.

## 7. CONCLUSIONS

We have presented an automated design method under droplet interference constraints for pin-constrained digital microfluidic biochips. We first investigated the interference problem for single droplets and between multiple droplets, and formulated the conditions for droplet-interference-free pin layouts for microfluidic arrays. A new pin-assignment method based on array partitioning has been proposed. Droplet interference is significantly reduced for pre-synthesis array partitioning and completely eliminated for post-synthesis array partitioning. In both scenarios (pre- and post-synthesis), we use a "Connect-5" pin-assignment algorithm to control the partitioned array using a small number of pins. The real-life example of a set of multiplexed bioassays has been used to evaluate

the effectiveness of the proposed method. By dramatically reducing the number of control pins with minimal impact on assay throughput, the proposed design method is expected to reduce cost and lead to the miniaturization of mixed-technology disposable biomedical devices for the emerging healthcare market. As part of ongoing work, we are developing an algorithm to automatically find optimal pin layouts with the maximum PCNI for a given number of control pins.

## REFERENCES


BREUER, M. A. 1977. A class of min-cut placement algorithms. In *Proceedings of the 14th Conference on Design Automation*. 284–290.

CHAKRABARYTY K. AND ZENG, J. 2005. Design automation for microfluidics-based biochips. *ACM J. Emerg. Tech. Comput. Syst. 1*, 186–223.

CHO, S. K., MOON, H. J., AND KIM, C. J. 2002. Toward digital microfluidic circuits: creating, transporting, cutting and merging liquid droplets by electrowetting-based actuation. In *Proceedings of IEEE MEMS Conference*. 32–52.

CONNECT5STRATEGIES, http://www.springfrog.com/games/gomoku/.

DIESTEL, R. 2005. *Graph Theory*. Springer, Berlin, Germany.

DUNLOP, A. AND KERNIGHAN, B. 1985. A procedure for placement of standard cell VLSI circuits. *IEEE Trans. CAD 1*, 92–98.

FAN, S. K., HASHI, C., AND KIM, C. J. 2003. Manipulation of multiple droplets on N×M grid by cross-reference EWOD driving scheme and pressure-contact packaging. In *Proceedings of IEEE MEMS Conference*. 694–697.

GONG, J. AND KIM, C. J. 2005. Two-dimensional digital microfluidic system by multi-layer printed circuit board. In *Proceeding of IEEE MEMS Conference*. 726–729.

KAHNG, A. B., MANDOIU, I. I., REDA, S., XU, X., AND ZEILIKOYSKY, A. 2003. Evaluation of placement techniques for DNA probe array layout. *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 262–269.

PAEGEL, B. M., BLAZEJ, R. G., AND MATHIES, R. A. 2003. Microfluidic devices for DNA sequencing: sample preparation and electrophoretic analysis. *Current Opinion Biotechnol 14*, 42–50.

PAPADIMITRIOU, C. H. 1993, *Computational Complexity*. Addison Wesley, Reading, MA.

POLLACK, M. G., FAIR, R. B., AND SHENDEROV, A. D. 2000. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Appl. Phys. Lett. 77*, 1725–1726.

SCHENA, M. 2000. *Microarray Biochip Technology*. Eaton Publishing, Natick, MA.

SCHULTE, T. H., BARDELL, R. L., AND WEIGL, B. H. 2002. Microfluidic technologies in clinical diagnostics. *Clinica Chimica Acta, 321*, 1–10.

SRINIVASAN, V., PAMULA, V. K., AND FAIR, R. B. 2004. An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids. *Lab on a Chip*, 310–315.

SRINIVASAN, V., PAMULA, V. K., PAIK, P., AND FAIR R. B. 2004. Protein stamping for MALDI mass spectrometry using an electrowetting-based microfluidic platform. In *Proceedings of SPIE*, vol. 5591, 26–32.

SU, F. AND CHAKRABARTY, K. 2004. Architectural-level synthesis of digital microfluidics-based biochips. In *Proceedings of ICCAD*. 223–228.

SU, F. AND CHAKRABARTY, K. 2005. Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. In *Proceedings of DAC*. 825–830.

SU, F. AND CHAKRABARTY, K. 2006. Module placement for fault-tolerant microfluidics-based biochips. *ACM Trans. Design Autom. Electr. Syst. 11*, 682–710.

SU, F., CHAKRABARTY, K. AND FAIR, R. B. 2006. Microfluidics-based biochips: technology issues, implementation platforms, and design automation challenges. *IEEE Trans. CAD, 25*, 211–223.

SU, F., WILLIAM, H., AND CHAKRABARTY, K. 2006. Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of Design, Automation and Test in Europe (DATE) Conference*. 323–328.

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

XU, T., SU, F., AND CHAKRABARTY, K. 2007. Defect-aware synthesis of droplet-based microfluidic biochips. *Proceedings of IEEE International Conference on VLSI Design*.

YUH, P.-H., YANG, C.-L., AND CHANG, Y.-W. 2006. Placement of digital microfluidic biochips using the T-tree formulation. In *Proceedings of DAC*. 931–934.

Received October 2006; revised February 2007; accepted February 2007