Module Placement for Fault-Tolerant Microfluidics-Based Biochips

FEI SU and KRISHNENDU CHAKRABARTY
Duke University

Microfluidics-based biochips are soon expected to revolutionize clinical diagnosis, DNA sequencing, and other laboratory procedures involving molecular biology. Most microfluidic biochips today are based on the principle of continuous fluid flow and they rely on permanently etched microchannels, micropumps, and microvalves. We focus here on the automated design of "digital" droplet-based microfluidic biochips. In contrast to conventional continuous-flow systems, digital microfluidics offers dynamic reconfigurability; groups of cells in a microfluidics array can be reconfigured to change their functionality during the concurrent execution of a set of bioassays. We present a simulated annealing-based technique for module placement in such biochips. The placement procedure not only addresses chip area, but also considers fault tolerance, which allows a microfluidic module to be relocated elsewhere in the system when a single cell is detected to be faulty. Simulation results are presented for case studies involving the polymerase chain reaction and multiplexed *in vitro* clinical diagnostics.

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 [**Computer Applications**]: Life and Medical Sciences—*Biology and genetics*; *health*

General Terms: Algorithms, Design, Performance, Reliability

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

1. INTRODUCTION

Microfluidics-based biochips are receiving considerable attention nowadays [Cho et al. 2002; Jones et al. 2001; Pollack et al. 2002; Verpoorte and De Rooij 2003]. These composite microsystems, which manipulate fluids on nanoliter-to-microliter scales, can greatly simplify cumbersome laboratory procedures. Such lab-on-a-chip devices are therefore expected to facilitate *in vitro* clinical diagnosis, DNA sequencing, and other common procedures in molecular biology.

This research was supported by the National Science Foundation under grant number IIS-0312352.
A preliminary and abridged version of this article was published in *Proceedings of the Design, Automation, and Test in Europe (DATE) Conference*, 2005, 1202–1207.

Authors' address: F. Su, K. Chakrabarty, Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708; email: {fs,krish}@ee.duke.edu.

Most microfluidic biochips today consist of permanently etched micropumps, microvalves, and microchannels, and are based on the principle of continuous fluid flow [Verpoorte and De Rooij 2003]. A promising alternative is to manipulate liquids as discrete microdroplets. This novel droplet-based approach is referred to in the literature as “digital microfluidics” [Cho et al. 2002; Pollack et al. 2002]. Each droplet can be controlled independently and each cell in the microfluidic array has the same structure. In contrast to continuous-flow systems, digital microfluidics offers dynamic reconfigurability as well as a scalable system architecture [Ding et al. 2001]. Groups of cells can dynamically be reconfigured to change their functionality during the execution of a bioassay. Multiple assays can be concurrently carried out on the microfluidic platform [Srinivasan et al. 2003].

The complexity of digital microfluidics-based biochips is expected to steadily increase due to the need for multiple and concurrent assays on the chip. Its time-to-market and fault tolerance are also expected to emerge as design considerations. As a result, current full-custom design techniques will not scale well for larger designs. There is a need to deliver the same level of CAD support to the biochip designer that is now available to the semiconductor industry. Moreover, it is expected that these microfluidic biochips will be integrated with microelectronic components in next-generation system-on-chip designs. The 2003 International technology roadmap for semiconductors (ITRS) clearly identifies the integration of electrochemical and electrobiological techniques as one of the system-level design challenges that will be faced beyond 2009, when feature sizes shrink below 50 nm [ITRS 2006].

Early research on CAD for digital microfluidics-based biochips was focused on device-level physical modeling of single components [Shapiro et al. 2003; Zeng and Korsmeyer 2004]. While top-down system-level design tools are now commonplace in IC design, few such efforts have been reported for digital microfluidic chips [Su and Chakrabarty 2004]. These biochip-specific synthesis tools can be used to relieve biochip users from the burden of manual optimization and implementation. Users will be able to describe bioassays at a sufficiently high level of abstraction; synthesis tools will then map the behavioral description to the microfluidic array and generate an optimized schedule of bioassay operations, the binding of assay operations to resources, and a layout of the microfluidic biochip. Thus, the biochip user can concentrate on developing the nano- and microscale bioassays, leaving implementation details to the synthesis tools.

We envisage the following steps in the synthesis of biochips. A behavioral model for a biochemical assay is first generated from the labotorary protocol for that assay. Next, architectural-level synthesis is used to generate a macroscopic structure of the biochip; this structure is analogous to a structural RTL model in electronic CAD. The macroscopic model provides an assignment of assay functions to biochip resources, as well as a mapping of assay functions to time-steps, based in part on the dependencies between them. Finally, geometry-level synthesis creates a physical representation at the geometrical level, that is, the final layout of the biochip consisting of the configuration of the microfluidic array, the locations of reservoirs and dispensing ports, and other geometric details. Based on previous work on architectural-level synthesis [Su and Chakrabarty 2004], we focus here on the problem of module placement for digital microfluidics-based biochips in a geometry-level synthesis phase.

The placement of microfluidic modules, such as different types of mixers and storage units, on a microfluidic array is a key problem in the design of digital microfluidics-based biochips. The ability to reconfigure the microfluidic array during the execution of bioassays makes this placement problem different from the traditional placement problem in electronic design. Moreover, many biochips are expected to be used for safety-critical applications, for example, patient health monitoring, neonatal care, and monitoring environmental toxins. Therefore, these biochips must be designed to be fault-tolerant such that they can continue to operate reliably in the presence of faults. One approach to fault tolerance is to carefully include spare cells in the array such that faulty cells can be bypassed without any loss of functionality. The locations of the spare cells must be determined by the physical design tool that maps modules to sets of cells in the array. We find that the placement of the microfluidic modules has a strong impact on the ease of reconfigurability for fault tolerance. Thus, in addition to area (measured by the number of cells in the array), fault tolerance is also a critical placement criterion.

In this article, we propose a design methodology that attempts to apply variants of classical module placement techniques to biochip design, with area and fault tolerance as the criteria. Two examples of real-life biochemical procedures, that is, polymerase chain reaction (PCR) and multiplexed *in vitro* diagnostics on human physiological fluids, are used to evaluate the proposed methodology. Since the placement problem is known to be NP-complete [Garey and Johnson 1979], a simulated annealing-based heuristic approach is developed to solve the problem in a computationally efficient manner. Solutions for the placement problem can provide the designer with guidelines on the size of the array to be manufactured. If module placement is carried out for a fabricated array, area minimization frees up more cells for sample collection and preparation. We also introduce a simple measure, referred to as the fault-tolerance index (FTI), to evaluate the fault-tolerance capability of the microfluidic biochip; this measure is incorporated into the placement procedure. This procedure leads to small biochip area due to the efficient utilization of dynamic reconfigurability, as well as high fault tolerance due to the efficient use of spare cells.

The organization of the remainder of this article is as follows. In Section 2, we present an overview of digital microfluidics-based biochips. Section 3 discusses related prior work. In Section 4, we present a simulated annealing-based heuristic for module placement in dynamically reconfigurable biochips. Next, in Section 5, the reconfiguration technique is studied in more detail and the fault-tolerance index (FTI) for single faults is defined. A fast algorithm to determine the FTI value is also presented. Furthermore, we show how the definition of the FTI can be extended to handle multiple faults. In Section 6, we incorporate the fault tolerance index into the placement procedure; we then use PCR and multiplexed diagnostics to evaluate the enhanced placement procedure. Finally, conclusions are drawn in Section 7.

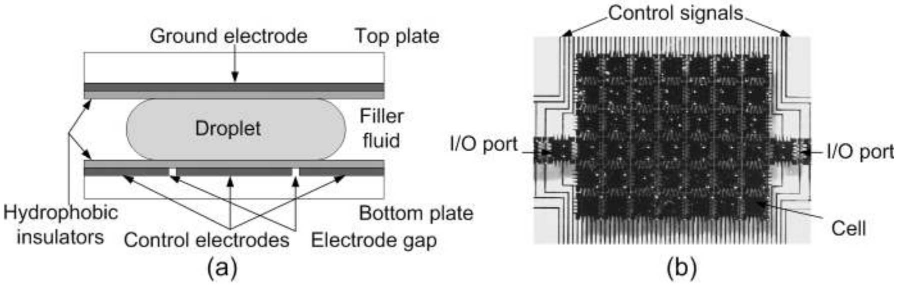


Fig. 1. (a) Basic cell used in a digital microfluidics-based biochip; (b) a 2-D array for digital microfluidics.

2. BACKGROUND

The operation of digital microfluidics-based biochips is based on the principle of electrowetting actuation. Electrowetting refers to the modulation of the interfacial tension between a conductive fluid and a solid electrode by applying an electric field between them. The basic cell of a digital microfluidics-based biochip is shown in Figure 1(a). The droplet containing biochemical samples, and a filler medium (such as silicone oil) are sandwiched between two parallel glass plates. The bottom plate contains a patterned array of individually controllable electrodes, while the top plate is coated with a ground electrode. A hydrophobic dielectric insulator is added to the plate to decrease the wettability of the surface and to add capacitance between the droplet and the control electrode. By varying the electrical potential along a linear array of electrodes, nanoliter-volume droplets can transport along this line of electrodes. The velocity of the droplet (up to 20cm/s) can be controlled by adjusting the control voltage (0∼90 V). Microdroplets can therefore be moved freely to any location of a two-dimensional array without the need for pumps and valves. Using a two-dimensional array, many common microfluidic operations for biomedical assays can be performed. For instance, the mixing operation is implemented by routing two droplets to the same location and then turning them around some pivotal points. Note that these operations can be performed anywhere on the array during the operation of the biochip, whereas in continuous-flow systems they must operate in a specific permanently etched micromixer or microchamber. This property of digital microfluidics-based biochips is referred to as dynamic reconfigurability, which we exploit here for high fault tolerance.

A module-based method, as in digital circuit design, can be applied to the design of digital microfluidic biochips. A microfluidic module library is provided to the designer after experimental characterization. This module library, analogous to the standard/custom cell library used in cell-based VLSI design, includes different microfluidic functional modules such as mixers and storage units. Each module is characterized by its function (mixing, storing, detection, etc.) and parameters such as width, length, and operation duration. During the biochip design, we can map the microfluidic assay operations to available microfluidic modules, and then use architectural-level synthesis techniques to determine a schedule of sets of bioassays subject to precedence constraints imposed by the corresponding assay protocols [Su and Chakrabarty 2004]. The locations of the modules on the microfluidic array are then determined by efficient placement algorithms. These configurations of the microfluidic array are loaded into a microcontroller that controls the voltages of electrodes in the array. Therefore, these modules can be dynamically formed by activating the corresponding control electrodes during run-time. In this sense, they can also be viewed as virtual devices.

3. RELATED PRIOR WORK

Physical design automation for integrated circuits, especially module placement, is a mature topic [Agnihotri et al. 2005; Cong et al. 2005; Sarrafzadeh and Wong 1996]. Heuristics, such as the TimberWolf placement method, based on simulated annealing are extensively used for custom/macrocell placement [Sechen 1988; Sechen and Sangiovanni-Vincentelli 1985], the placement problem is often formulated as 2-D rectangle packing [Sechen and Sangiovanni-Vincentelli 1985]. Many techniques for 2-D placement/floorplanning, for example, methods based on the sequence pair and transitive closure graph (TCG), have been published in the literature [Lin and Chang 2001; Murata et al. 1995]. Since these techniques do not consider reconfigurability, they are not directly applicable to programmable devices. Dynamically reconfigurable FPGAs (DRFPGAs) have received much attention recently [Bazargan et al. 2000a; 2000b; Cong et al. 2005; Comg and Lim 2004]. Several recent methods, such as the 3D-subTCG or sequence triplet [Yuh et al. 2004a; 2004b], have been derived from well-known 2-D placement algorithms to handle placement/floorplanning for reconfigurable devices. Note that the partial reconfiguration offered by DRFPGAs is in many ways similar to the dynamic reconfigurability provided by digital microfluidics-based biochips. However, to the best of our knowledge, no placement techniques reported thus far for DRFPGAs have taken fault tolerance into account. Fault tolerance/reliability is a critical design metric for most digital microfluidic biochips. Thus, a new module placement method is needed for fault-tolerant biochip designs.

Moreover, the programmability of DRFPGAs is limited by the well-defined roles of interconnect and logic blocks. Interconnect cannot be used for storing information, and logic blocks cannot be used for routing. In contrast, digital microfluidics-based biochips offer significantly more programmability. The cells in the microfluidic array can be used for storage and functional operations, as well as for transporting fluid droplets.

As integrated circuits become denser, reliability emerges as a major challenge. Historically, reliability has been addressed through robust manufacturing processes. However, this approach does not address the reliability issues associated with system design. In recent years, design-for-reliability (DFR) methodologies have been incorporated into the chip design flow and into CAD tools to address the challenge posed by deep submicron techniques [Yang and Chern 1993]. Although microelectromechanical systems (MEMS) is a relatively young field compared to integrated circuits, reliability studies for MEMS have received considerable attention [McCluskey 2002]. However, due to significant differences in the actuation principles underlying digital microfluidics and MEMS, these reliability enhancement techniques cannot be directly used for the design of microfluidics-based biochips. Recently, fault analysis and test methodology have been developed for digital microfluidics-based biochips [Su et al. 2003]. This cost-effective test methodology facilitates online testing, which allows fault testing and biochemical assays to run simultaneously on a microfluidic biochip [Su et al. 2004]. It also facilities fault tolerance of digital microfluidics-based biochips.

While MEMS design tools have reached a certain level of maturity [Mukherjee and Fedder 1998], CAD tools for biochips are still in their infancy. Design automation techniques have been proposed for DNA probe arrays that are used for hybridization and DNA sequencing [Andronescu et al. 2003; Bradley and Skiena 1997; Kahng et al. 2005]. However, the digital microfluidic biochips described in this article are more versatile and complex than DNA arrays. Current design methodologies for microfluidics-based biochips are typically full-custom and bottom-up in nature. Since much of the microfluidics work to date has been focused on device development, most design automation research for microfluidic biochips has been limited to device-level physical modeling of components [Chatterjee and Aluru 2005; Wang et al. 2005; White 2004; Zeng and Korsmeyer 2004]. In addition, some commercial computational fluidic dynamics (CFD) tools, such as CFD-ACE+ from CFD Research Corporation and FlumeCAD from Coventor, Inc., support 3-D simulation of fluidic transport. A recent release of CoventorWare from Coventor, Inc., includes microfluidic behavioral models to support system-level design. Pfeiffer et al. [2005] also presented a synthesis approach for multiplexed capillary electrophoresis (CE) separation microchips. Unfortunately, these CAD tools are only able to deal with continuous-flow systems, and they are therefore inadequate for the design of digital microfluidics-based biochips. Recently, both behavioral performance analysis and modeling for droplet-based microfluidic systems have been investigated [Böhringer 2005; Ding et al. 2001; Griffith and Akella 2005; Yuh et al. 2004]. As a first effort, Su and Chakrabarty [2004] introduced architectural-level synthesis to digital microfluidic biochip design. An integer linear programming (ILP) model and classical high-level synthesis techniques (e.g., list scheduling) are applied to optimally schedule the bioassay operations and also to assign them to appropriate modules. This prior work serves as a basis for the placement method presented in this article. We assume here that the droplet transportation time between different modules (e.g., 5–10 ms) is negligible compared to assay operation times (e.g., 5–10 s for mixing), and that droplet pathways can be trivially determined. Extensions to this article will include module placement strategies that do not make these assumptions.

4. MODULE PLACEMENT

Placement is one of the key physical design problems for digital microfluidics-based biochips. Based on the results obtained from architectural-level synthesis (i.e., a schedule of bioassay operation, a set of microfluidic modules, and the binding of bioassay operations to modules) placement determines the locations

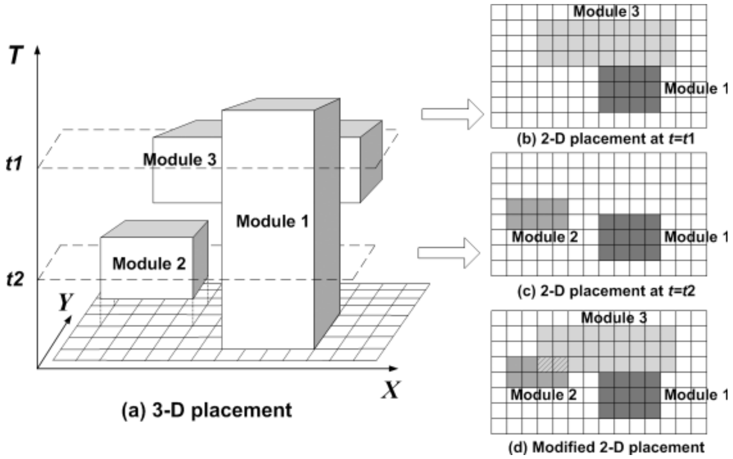

T
t1
t2
Y
X
Module 3
Module 1
Module 2
(a) 3-D placement
(b) 2-D placement at t=t1
(c) 2-D placement at t=t2
(d) Modified 2-D placement

Fig. 2. Reduction from 3-D placement to a modified 2-D placement.

of each module on the microfluidic array in order to optimize some design metrics. Since digital microfluidics-based biochips enable dynamic reconfiguration of the microfluidic array during run-time, they allow the placement of different modules on the same location during different time intervals. Thus, the placement of modules on the microfluidic array can be modeled as a 3-D packing problem. Each microfluidic module is represented by a 3-D box, the base of which denotes the rectangular area of the module and the height denoting the time-span of its operation. Microfluidic biochip placement can now be viewed as the problem of packing these boxes to minimize the total base area while avoiding overlaps.

Since placement follows architectural-level synthesis in the proposed synthesis flow, the starting times for each operation corresponding to a module, that is, their positions in the time axis, are predetermined. Therefore, the 3-D packing problem can be reduced to a modified 2-D placement problem. The horizontal cuts with the 3-D boxes correspond to the configurations of the microfluidic array at different points in time. For example, in Figure 2, the cut $t = t_1$ corresponds to the 2-D placement shown in Figure 2(b), and the cut $t = t_2$ corresponds to another configuration in Figure 2(c). The configurations of the microfluidic array during different time intervals can be combined together to form the modified 2-D placement shown in Figure 2(c). Note that the base of the 3-D box representing module i should be placed on the cutting plane $t = S_i$, where S_i is the starting time of module i's operation as determined by architectural-level synthesis. The modules can arbitrarily slide on these fixed cutting planes while avoiding overlap. Thus, instead of a 3-D packing problem, we only need to consider a modified 2-D placement consisting of several 2-D configurations in different time spans.

The module placement problem for electronic design is known to be NP-complete [Garey and Johnson 1979]. The microfluidic placement problem can also be shown, by the method of restriction, to be NP-complete. Consequently, heuristics are needed to solve the placement problem in a computationally efficient manner. Simulated annealing is a well-studied combinatorial optimization method, and it has been extensively used for traditional module placement problems [Cassoto et al. 1987; Chandy et al. 1997; Sechen 1988; Sechen and Sangiovanni-Vincentelli 1985]. An advantage of simulated annealing is that it explores the configuration space of the optimization problem while allowing *hill-climbing* moves, that is, the acceptance of new configurations that increase the cost. In this article, we develop a simulated annealing-based algorithm to solve the placement problem for digital microfluidics-based biochips.

There are two different ways to solve placement problems using simulated annealing: the direct approach and the indirect approach. In the direct approach, the annealing procedure is applied directly to the actual physical coordinates, sizes, and orientations of the modules. This approach cannot guarantee that each new placement during the annealing procedure is a feasible solution without any forbidden overlap. Thus, the penalty for such forbidden overlaps must be included in the cost function. Its main advantage is that it has the detailed geometrical information of a layout configuration which facilitates evaluation of the associated fault-tolerance capability. On the other hand, in the indirect approach, the simulated annealing algorithm is based on an abstract representation (i.e., code) of the placement. The coding process usually obtains a graph representation or a module sequence to describe topological relationships between modules. Then a subsequent mapping (i.e., decoding) process is used to generate a placement from its corresponding code. There are many floorplan/placement coding methods proposed in the literature for 2-D placement, for example, sequence pair (SP) [Murata et al. 1995], B*-tree [Chang et al. 2000], transitive-closure graph (TCG) [Lin and Chang 2001], O-tree [Guo et al. 1999], twin binary sequence (TBS) [Young et al. 2003], and Q-sequence [Sakanushi and Kajitani 2000]. In recent years, several methods have been proposed for 3-D placement in reconfiguration computing, for example, sequence triplet, 3-D sub-TCG and T-tree [Yuh et al. 2004a; 2004b]. These methods also belong to the indirect approach category. The advantage of these approaches is that all intermediate placement solutions are feasible, thus, they optimize the search for solution space. However, as will be shown in the next section, evaluating the fault-tolerance capability for digital microfluidic biochips requires the actual physical information of a placement configuration. The indirect approaches are, unfortunately, inefficient in dealing with fault tolerance issues for biochip placement. Thus, instead of using a complicated problem-encoding scheme, we employ the direct approach for our placement problem. This simulated annealing-based algorithm seeks to optimize the design metric (e.g., biochip area or fault-tolerance) while driving the overlap penalty to zero. Some important details of the proposed algorithm are as follows.

(a) *Initial Placement*: It has been reported in the literature that the initial configuration has little impact on the final outcome of a simulated annealing-based optimization [Sechen 1988]. Therefore, we apply a simple constructive approach to a formulate the initial placement, as shown in Figure 3. In addition,

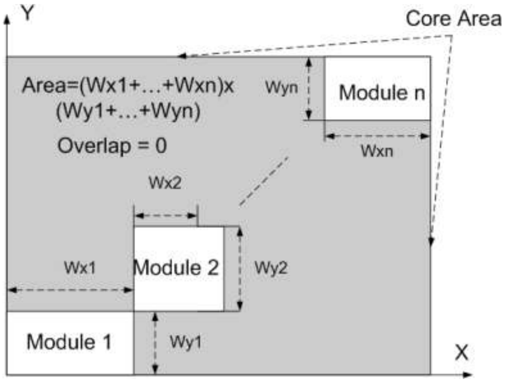


Fig. 3. Initial placement in the simulated annealing procedure.

during the annealing process the modules are prevented from being placed outside the boundaries of the core area, as defined by Figure 3.

(b) *Generation Function*: New placements can be generated in several ways: (i) a single microfluidic module is randomly selected to be moved to a randomly-chosen location; (ii) a single module is randomly displaced to a new location and the orientation of this module is changed; (iii) a pair of modules are randomly selected for interchange; and (iv) a pair of modules are interchanged, in which at least one module has its orientation changed. During the annealing process, we assign the probability p to a single-module displacement and $1-p$ to a two-module interchange. An effective ratio of $p/(1-p)$ is determined experimentally.

(c) *Controlling Window for Single-Module Displacement*: The displacement of a single module by a large distance leads to a large increase in the cost metric ($\Delta C > 0$). At low temperatures, during the annealing process, only the new generations with $\Delta C \leq 0$ have a reasonable chance of being accepted. This increases the probability that displacements over large distances are rejected. We apply a controlling window to discourage long-distance displacements at low temperatures. As the temperature approaches zero, the span of the controlling window reaches its minimum value; this condition is used as the stopping criterion for simulated annealing.

(d) *Annealing Scheme*: Most annealing parameters are experimentally determined. These include the following: (i) the temperature is modulated as $T_{new} = \alpha \times T_{old}$, where α is a cooling factor (e.g., $\alpha = 0.9$); (ii) the number of iterations of the inner loop for a given value of T is determined using the relationship $N = Na \times Nm$, where Na is a constant number (e.g., $Na = 400$), and Nm is the number of the modules; (ii) the initial temperature T_∞ is chosen to ensure that almost every new placement can be accepted. For example, we set $T_\infty = 10000$.

(e) *Cost Metrics*: Cost metrics are used to mathematically represent the optimization goals of the placement problem. We consider the area of the array and the degree of fault-tolerance as cost metrics. As an important cost metric, the fault-tolerance capability of a biochip placement is discussed in the next section.

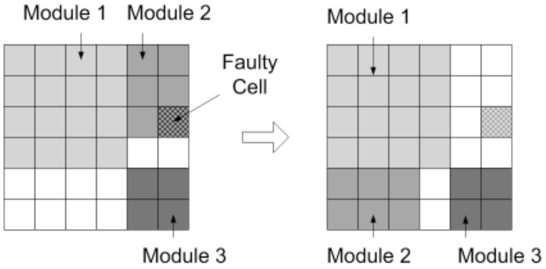


Fig. 4. Example of partial reconfiguration.

5. FAULT-TOLERANCE AND RECONFIGURATION

In this section, we investigate dynamic and partial reconfiguration to avoid a faulty cell in the microfluidic array. Based on this reconfiguration technique, a simple numerical measure termed the fault-tolerance index is defined to estimate the fault-tolerance capability of the biochip. We also present an efficient algorithm to determine the fault-tolerance index of a biochip configuration based on the notion of maximal-empty rectangles. We further extend the definition of this index to handle multiple faults.

5.1 Partial Reconfiguration

A digital microfluidics-based biochip can be viewed as a dynamically reconfigurable system. If a cell becomes faulty during the operation of the biochip, it will be detected using the technique described in Su et al.[2003, 2004]. The microfluidic module containing this cell can easily be relocated to another part of the microfluidic array by changing the control voltages applied to the corresponding electrodes. An example of partial reconfiguration is shown in Figure 4. Fault-free unused cells in the array are utilized to accommodate the faulty module. Hence, the configuration of the microfluidic array, that is, the placement of the microfluidic modules, influences the fault-tolerance capability of the biochip. Moreover, since partial reconfiguration only targets the module containing the faulty cell and leaves other aspects of the microfluidic configuration unchanged, a fast heuristic algorithm can be used to find a new location for this module. Therefore, partial reconfiguration is suitable for dynamic online reconfiguration during field operation of the microfluidic biochip.

5.2 Fault-Tolerance Index

In order to facilitate partial reconfiguration and incorporate fault tolerance in the simulated annealing-based placement procedure, we need to evaluate the fault-tolerance capability of the microfluidic biochip.

We consider the reconfiguration problem for a single failing cell in the microfluidic array. The single fault assumption is valid when testing and reconfiguration are carried out frequently and the “abort-at-first-fail” test strategy is applied. We also assume that every cell has the same failure probability. Since microfluidic biochips have not yet been manufactured in large numbers, failure data or statistical models are not readily available. Due to the fact that each cell has the same basic structure, the assumption of uniform failure probability is reasonable for digital microfluidics-based biochips. The failure model can be easily updated when statistical failure data becomes available.

We use a 2-D coordinate system to refer to the cells in the microfluidic array. The bottom-left cell is referred to as (1, 1) and the top-right cell in an $m \times n$ array is referred to as (m, n). For an $m \times n$ microfluidic array, assume that an arbitrary cell (i, j) is faulty. For a given microfluidic configuration C, if this cell is contained in a module, we attempt to apply partial reconfiguration to relocate this module to avoid the faulty cell. If this reconfiguration succeeds, that is, if we find an adequate number of contiguous cells to accommodate this module, or if cell (i, j) is not used by any module, we deem this cell to be *C-Covered* for this configuration. Otherwise, cell (i, j) is not C-Covered. For an array with k C-Covered cells, we define the fault-tolerance index (FTI) as follows: $\text{FTI} = k/(m \times n)$.

Note that the FTI lies between 0 and 1. It increases if there are more C-Covered cells in the array. If the FTI is 1, it implies that when any single cell in the array is faulty, this microfluidic configuration can be used by applying partial reconfiguration to bypass the faulty cell. Obviously, this type of design has high reliability and good fault-tolerance capability. On the other hand, if the FTI is 0, the biochip cannot be reconfigured if any arbitrary cell becomes faulty. This is the worst case scenario and needs to be avoided.

In order to determine if a cell is C-Covered for configuration C, we use an efficient procedure based on the notion of maximal-empty rectangles. The details of this procedure are described now.

5.3 Fast Algorithm to Determine the FTI

Our goal is to find maximal-empty rectangles in the microfluidic array, and then check if these rectangles can accommodate the faulty module. A maximal empty rectangle (MER) is defined as an empty rectangle (a set of unused cells) that cannot be completely covered by any other empty rectangles. If a maximal-empty rectangle can accommodate the faulty module, this module can be relocated to the empty rectangle to avoid the faulty cell. If no such maximal-empty rectangle exists, partial reconfiguration is deemed to have failed. We then conclude that the corresponding faulty cell is not C-Covered.

An encoding method is first used to facilitate the implementation of this algorithm. If a module contains a faulty cell, this module is temporarily removed from the placement. Next, the configuration of the microfluidic array is modeled by a matrix consisting of 0s and 1s. The faulty cell and all cells contained in the currently operational modules are represented by 1s; all unused cells are represented by 0s. Such an encoding example is shown in Figure 5. Note that each empty rectangle should only consist of 0 cells.

In order to find all maximal-empty rectangles rapidly, a data structure referred to as the staircase [Edmonds et al. 2003; Handa and Vemuri 2004], is employed in the algorithm. A staircase (x, y) is defined as the collection of all overlapping empty rectangles with (x, y) as their bottom-right corner. Figure 6

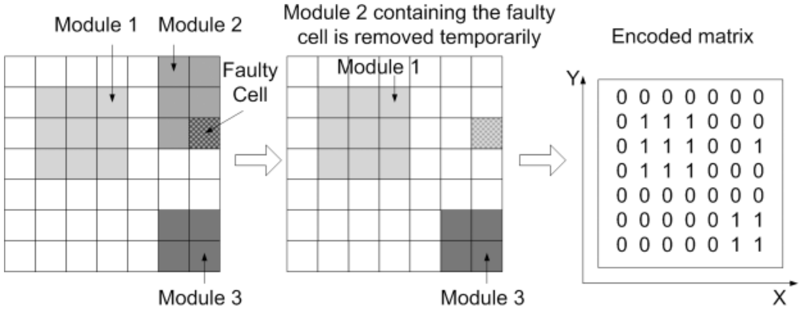


Fig. 5. An example of encoding a microfluidic array.

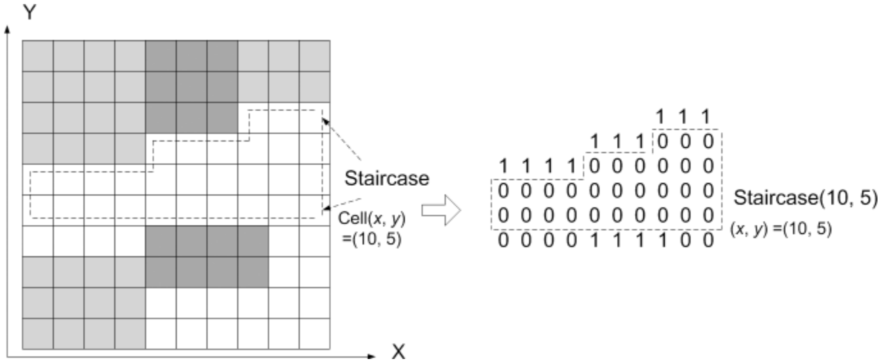
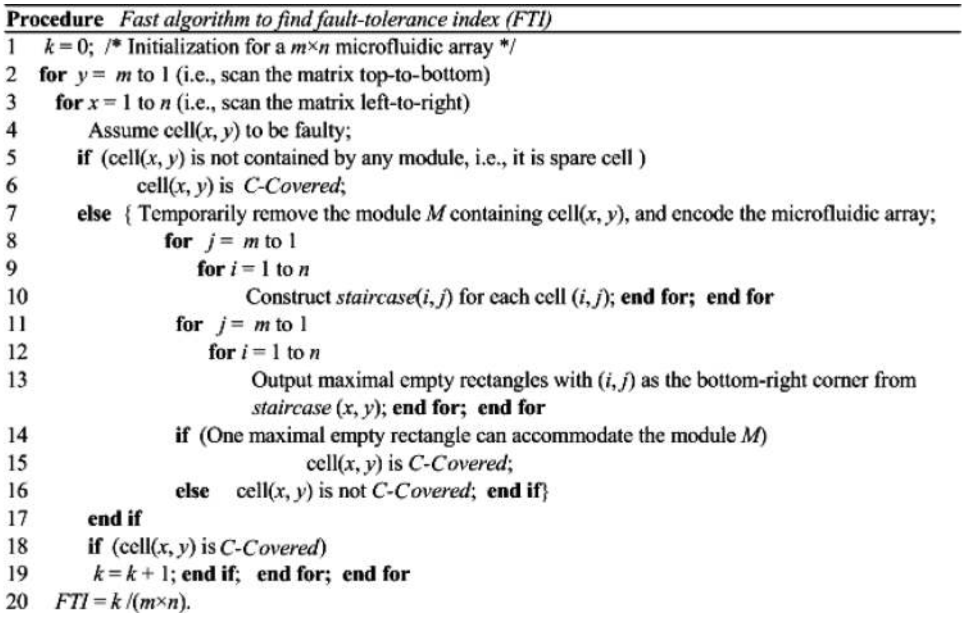


Fig. 6. An example of staircase representation.

shows an example of a staircase in the microfluidic array and the corresponding encoded matrix.

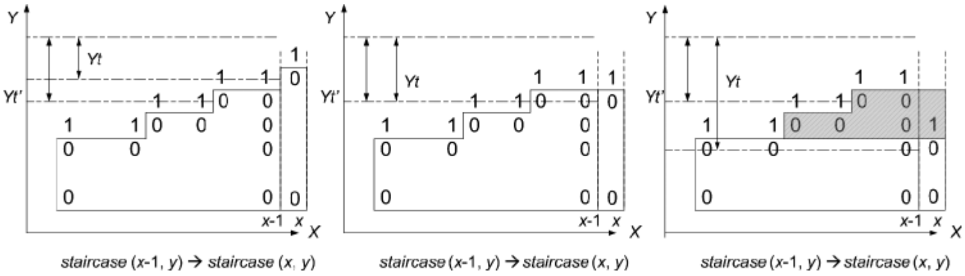
The data structure staircase (x, y) helps to determine all maximal-empty rectangles that lie entirely within staircase (x, y) whose bottom-right corner is (x, y). The algorithm traverses the matrix left-to-right and top-to-bottom, creating a staircase for every cell in the matrix. Next, based on knowledge of staircases, all maximal-empty rectangles are determined. An algorithmic structure for constructing staircases and generating maximal-empty rectangles from staircases is shown in Figure 7. Some important details are described as follows.

(a) *Constructing Staircase* (x, y)
Since a staircase is only possible at an empty location, staircase $(x, y) = null$ if cell (x, y) is represented by 1. Otherwise staircase (x, y) can be easily constructed from a previous staircase at point $(x - 1, y)$, as shown in Figure 8 [Edmonds et al. 2003]. We define Yt as the Y-coordinate of the top-most 0-cell in the block of empty cells in column x starting at cell (x, y), and similarly Yt' is defined for cell $(x - 1, y)$. Staircase (x, y) is constructed from staircase $(x - 1, y)$, depending on the comparison between Yt and Yt'. There are three different cases: If $Yt > Yt'$, staircase (x, y) is constructed simply by adding an additional empty column starting from cell (x, y) to cell (x, Yt). If $Yt = Yt'$, staircase (x, y) is the same as the old staircase $(x - 1, y)$ except that the top step



```
Procedure  Fast algorithm to find fault-tolerance index (FTI)
k = 0; /* Initialization for a $m \times n$ microfluidic array */
for y = m to 1 (i.e., scan the matrix top-to-bottom)
  for x = 1 to n (i.e., scan the matrix left-to-right)
    Assume cell(x, y) to be faulty;
    if (cell(x, y) is not contained by any module, i.e., it is spare cell )
        cell(x, y) is C-Covered;
    else { Temporarily remove the module M containing cell(x, y), and encode the microfluidic array;
          for j = m to 1
            for i = 1 to n
              Construct staircase(i, j) for each cell (i, j); end for; end for
          for j = m to 1
            for i = 1 to n
              Output maximal empty rectangles with (i, j) as the bottom-right corner from
              staircase (x, y); end for; end for
          if (One maximal empty rectangle can accommodate the module M)
                cell(x, y) is C-Covered;
          else   cell(x, y) is not C-Covered; end if}
    end if
    if (cell(x, y) is C-Covered)
     k = k + 1; end if; end for; end for
FTI = k /(m×n).
```

Fig. 7. Pseudocode for algorithm to obtain the fault-tolerance index.

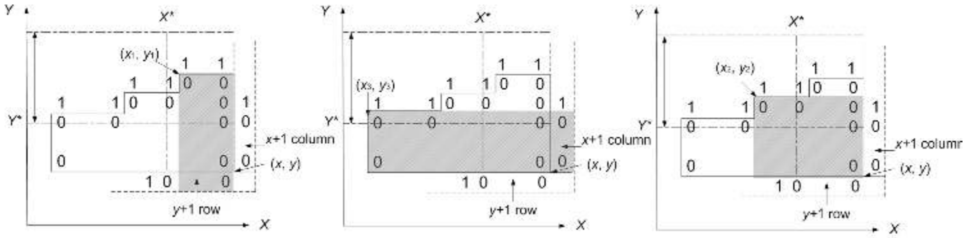


staircase (x-1, y) → staircase (x, y) staircase (x-1, y) → staircase (x, y) staircase (x-1, y) → staircase (x, y)

Fig. 8. Illustration of constructing staircase $(x,\ y)$ from staircase $(x - 1,\ y)$.

is extended one column to the right. If $Yt < Yt'$, staircase $(x,\ y)$ is constructed from staircase $(x - 1,\ y)$ by chopping off the empty area with the Y-coordinate larger than Yt.

(b) *Generating the Maximal-Empty Rectangles from Staircases*
The rectangle contained in staircase $(x,\ y)$ is considered a maximal empty rectangle if this rectangle cannot be extended further. We define Y^* as the Y-coordinate of the top-most 0-cell in the block of empty cells in column $x + 1$ starting at cell $(x + 1,\ y)$, and X^* as the X coordinate of the left-most 0-cell in the block of empty cells in row $y + 1$ starting at cell $(x,\ y + 1)$. Consider an empty rectangle with the top-left corner $(x_i,\ y_i)$, that is, one step of staircase $(x,\ y)$, and the bottom-right corner $(x,\ y)$. If $x_i \geq X^*$, this rectangle can be extended one more row down. If $y_i \leq Y^*$, this rectangle can be extended one more column to the right. Therefore, this empty rectangle is maximal if and only if $x_i < X^*$ and $y_i > Y^*$; as shown in Figure 9. With such a checking process,



When $x_1 > x^*$, rectangle (x_1, y_1, x, y) can be extended down further

When $y_3 = y^*$, rectangle (x_3, y_3, x, y) can be extended to the right further

When $x_2 < x^*$ and $y_2 > y^*$, rectangle (x_2, y_2, x, y) is maximal-empty rectangle

Fig. 9. Example of generating maximal-empty rectangles from staircases.

all maximal empty rectangles with a bottom-right corner (x, y) will be found. Scan each cell in the array, and then all maximal empty rectangles will finally be obtained.

With the previous efficient algorithm, whose computational complexity is $O((m \times n)^2)$ for an $m \times n$ microfluidic array, we can easily incorporate the FIT into the design metric of the simulated annealing-based algorithm presented in Section 4. This enhanced placement algorithm targets at high reliability of biochips; it will be described with details in Section 6.

5.4 Extending the FTI to Multiple Faults

As stated before, the FIT definition and the algorithm used to compute it in Sections 5.2 and 5.3 are based on a single fault assumption. However, since multiple defects are likely in a fabricated microarray, multiple faulty cells are more likely than a single failing cell. Thus, we need to evaluate the fault-tolerance capability of a biochip design for multiple faults. In this section, we extend the definition of the FTI to handle this problem.

First, we introduce a new defect model that statistically describes the spatial distribution of defective cells in the microfluidic array. In this model, we assume that each cell of a microfluidic array has the same defect probability q. Moreover, the failures of the cells are independent of each other. We refer to $p = 1 - q$ as the survival probability of a single cell. Note that the assumption of equal survival probabilities is reasonable, since each cell in the microfluidic array has the same structure. In addition, the assumption of independent failures is valid for random and small spot defects which result from imperfect materials and from undesirable chemical and airborne particles. Compared to the previous single fault assumption, the assumptions regarding multiple faulty cells complicate the fault-tolerance and reconfiguration technique, but they make the defect model more realistic.

Based on these assumptions, we extend the FTI definition as follows: for a 2-D microfluidic array, its fault-tolerance index (FTI) is defined as the probability that fault-tolerance can be achieved via successful partial reconfiguration when the array contains one or multiple faulty cells. Note that the FTI definition in Section 5.2 can be viewed as a special case when only one cell is faulty and the failure probability is the same for all cells.

With the new definition of FTI, we now need an efficient way to determine the reconfiguration probability for a given microfluidic array configuration (i.e., module placement). Monte-Carlo simulation is a popular method for probabilistic analysis, and it can be used to estimate the FTI value. During each run of the simulation, the cells in the microfluidic array are randomly chosen to fail with a probability defined by the above multifault model. We then attempt to tolerate these defects through partial reconfiguration. By generating a very large number of simulation runs, the FTI can be determined by the percentage of successful reconfigurations. This approach is simple and straightforward; however, its inherently high computational complexity precludes its use in our simulated annealing-based placement algorithm. Since we need to evaluate FTI for each intermediate biochip placement during the annealing process, a more efficient FTI estimation method is needed.

Here, we propose a new method to quickly estimate the value of the FTI. The key idea underlying this estimation method is to restate the problem of finding the reconfiguration probability of a biochip configuration in terms of the problem of estimating the survival probability for each microfluidic module in the array. Based on the independence assumption, the FTI value for configuration C, that is, FTI(C), can then be easily estimated by multiplying the survival probabilities of all the modules, as follows:

$$\begin{aligned}\mathrm{FTI}(C) &\approx \Pi P_s(M_i)\\ &= \Pi[1-f_1(M_i)+f_1(M_i)\times f_2(M_i)],\end{aligned}$$

where M_i, $i = 1\cdots N$, is the microfluidic module (e.g., mixer) contained in a given microfluidic configuration C, $P_s(M_i)$ is the survival probability of module M_i, $f_1(M_i)$ is the probability of the module M_i being faulty, and $f_2(M_i)$ is the probability that M_i can be successfully reconfigured if it becomes faulty. Note that the survival probability of each module consists of two parts: the first part is its probability of being fault-free, and the second part is the probability of successful reconfiguration. Here, the multiplication of survival probabilities of microfluidic modules provides an estimate of the FTI value. By calculating the f_2 values of the modules independently, the probability of successful reconfiguration is calculated optimistically. Since the FTI is only used to guide the placement optimization procedure, the estimate is useful during module placement.

For each microfluidic module M_i, it is easy to determine the value of $f_1(M_i)$ based on the aforementioned multifault assumption, that is, $f_1(M_i) = 1-p^{A(Mi)}$, where $A(M_i)$ is the total number of cells contained in M_i. It is obvious that $f_1(M_i)$ increases with $A(M_i)$. Yet it is not trivial to determine the value of $f_2(M_i)$. Instead of invoking complicated procedures involving fault simulation, we simply examine the biochip configuration (e.g., its empty spaces) and estimate the ease of reconfiguration. For example, partial reconfiguration for module M_i is easier if the maximum empty rectangle (MER) for M_i is relatively large. Thus, $f_2(M_i)$ increases with the ratio of MER size to the area of M_i, that is, with $R(M_i) = A(\mathrm{MER})/A(M_i)$. Thus, we can estimate the value of $f_2(M_i)$ using a simple function of $R(M_i)$ and other variables. For example, $f_2(M_i) \approx 1 - (1 - p^{A(Mi)})^{R(Mi)}$, whereby we divide the MER for M_i

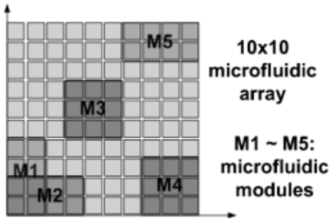


Fig. 10. Biochip placement example used to evaluate the proposed FTI estimation method.

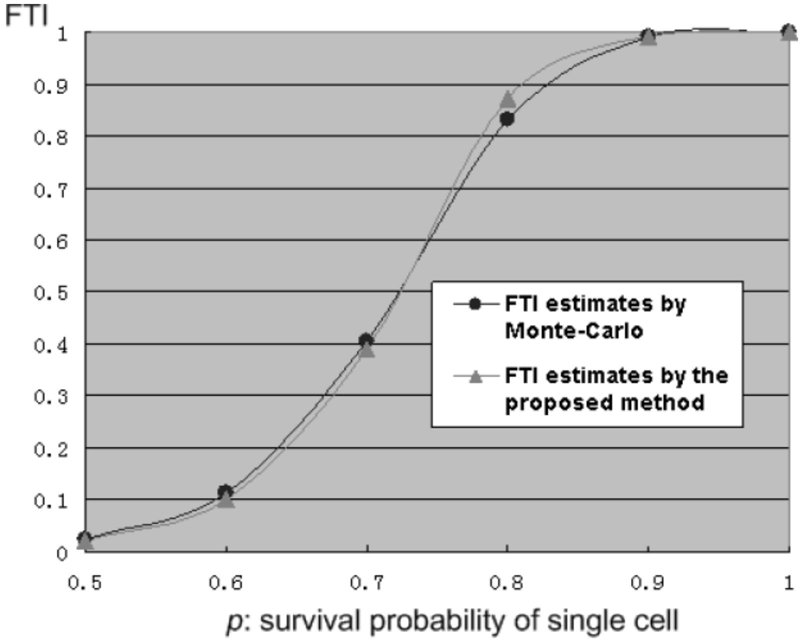


Fig. 11. An evaluation of the FTI estimate for multiple faults.

into $R(M_i)$ clusters, and then the reconfiguration probability $f_2(M_i)$ for M_i can be determined by the likelihood of having at least one fault-free empty cluster. We can further include some constants into the previous function: $f_2(M_i) \approx 1 - k_1(1 - (k_2 p)^{k3A(Mi)})^{R(Mi)}$, where constants $k_1 \sim k_3$ can be fine-tuned through experiments.

In this way, we can easily estimate the FTI value by studying the array configuration properties, especially microfluidic module areas and the associated empty spaces. Note that this method can also be applied to the estimation of the FTI in the case of a single fault. Although only an estimate is obtained, the estimate is an efficient way to calculate the FTI value without sweeping the whole array, as described in Section 5.3.

In order to evaluate its effectiveness, we now apply the proposed FTI estimation method to the biochip placement example shown in Figure 10. There are five modules contained in the array. Note that modules M1 and M2 share cells during different time-spans. Figure 11 shows the FTI estimates for different

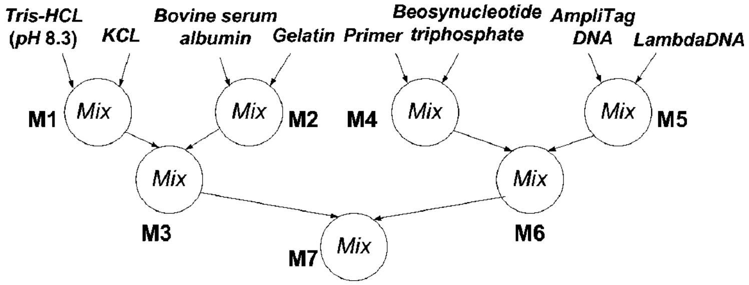


Fig. 12. Sequencing graph for the mixing stage of PCR.

values of p, and also compares them to the results obtained using Monte-Carlo simulation. We set $k_1 = 0.75, k_2 = 0.45$, and $k_3 = 1.15$ to determine $f_2(M_i)$, and we used 5000 simulation runs in Monte-Carlo simulation. Experiments show that the FTI values estimated by the proposed method are very close to the results obtained using Monte-Carlo simulations. While the calculation of the FTI by Monte-Carlo simulation for each value of p takes approximately seven minutes of CPU time, our proposed estimation method takes less than one second. Note that all the CPU times reported in this article are for a 1.0 GHz Pentium-III PC with 256 MB of RAM.

6. EXPERIMENTAL EVALUATION

In this section, we present an enhanced placement algorithm with the biochip array area and the fault-tolerance capability as design metrics. Two typical real-life biochemical applications of digital microfluidic biochips, namely polymerase chain reaction and multiplexed *in vitro* clinical diagnostics, are used to illustrate and evaluate the proposed methods.[1]

6.1 Example 1: PCR

Polymerase chain reaction (PCR) is one of the most common techniques for DNA analysis [Meltzer 1998]. It is used for rapid enzymatic amplification of specific DNA fragments. PCR can amplify genomic DNA exponentially, using temperature cycles. Recently, the feasibility of performing droplet-based PCR on digital microfluidics-based biochips has been successfully demonstrated [Srinivasan et al. 2003]. Here, we use the mixing stage of PCR as a first example to evaluate the simulated annealing-based placement algorithm that facilitates fault tolerance. Its assay protocol can be modeled by a sequencing graph [Zhang et al. 2002], as shown in Figure 12.

Based on this graph model, architectural-level synthesis can be used to carry out both resource binding and scheduling. Let the resource binding be as shown in Table I. Note that the module generated here has a segregation region wrapped around the functional region, which not only isolates the functional region from its neighbors but also provides a communication path

[1] These examples for digital microfluidic biochip design and synthesis are available as a "benchmark" on the web at http://www.ee.duke.edu/~fs/Benchmark.pdf.

Table I. Resource Binding in PCR

Operation	Hardware*	Module	Mixing Time
M1	2 × 2 electrode array	4 × 4 cells	10 s
M2	4-electrode linear array	3 × 6 cells	5 s
M3	2 × 3 electrode array	4 × 5 cells	6 s
M4	4-electrode linear array	3 × 6 cells	5 s
M5	4-electrode linear array	3 × 6 cells	5 s
M6	2 × 2 electrode array	4 × 4 cells	10 s
M7	2 × 4 electrode array	4 × 6 cells	3 s

*electrode pitch: 1.5 mm; gap height: 600 μm.

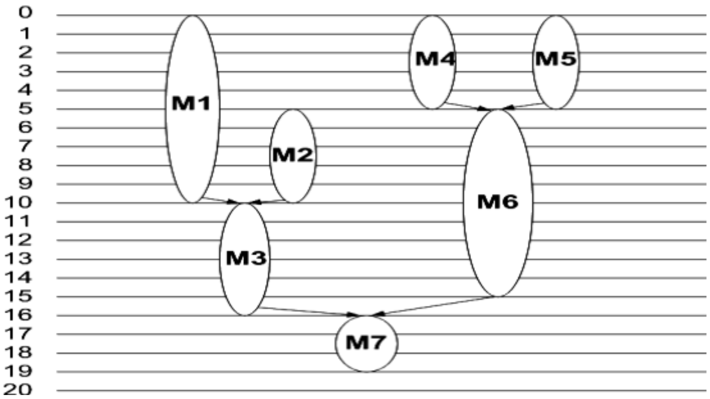


Fig. 13. Schedule highlighting the usage of microfluidic modules.

for droplet movement. The data for the operation times associated with the different modules is obtained from real-life experiments [Paik et al. 2003]. A schedule for the functional operations and module usage is shown in Figure 13.

First we apply a sequence pair (SP)-based method to this example as a baseline for assessing the quality of the proposed placement method. The well-known sequence pair algorithm is very popular for handling 2-D floorplanning/placement [Murata et al. 1995]. A sequence pair is a succinct representation of nonslicing floorplans of rectangles which consists of the two module sequences (Γ_+, Γ_-). The geometrical relationship between two modules is defined as follows: (1) module M_1 is right (/left) to module M_2 if and only if M_1 is after (/before) M_2 in both Γ_+ and Γ_-; and (2) M_1 is above(/below) module M_2 if and only if M_1 is before(/after) M_2 in Γ_+ and after (/before) M_2 in Γ_-. A given sequence pair can be evaluated (i.e., translated to its corresponding block placement) through the use of horizontal and vertical constraint graphs [Murata et al. 1995]. Recently, an enhanced algorithm has been proposed in Tang et al. [2001], and Tang and Wong [2001] to improve the speed of sequence pair evaluation. Instead of constructing horizontal and vertical constraint graphs, the algorithm is based on the computation of the longest common subsequence in a pair of weighted sequences. The enhanced method can examine more sequence pairs, thereby leading to a better placement solution in less run-time. The sequence pair-based placement algorithm also uses simulated annealing, whereby the generation and evaluation of a large number

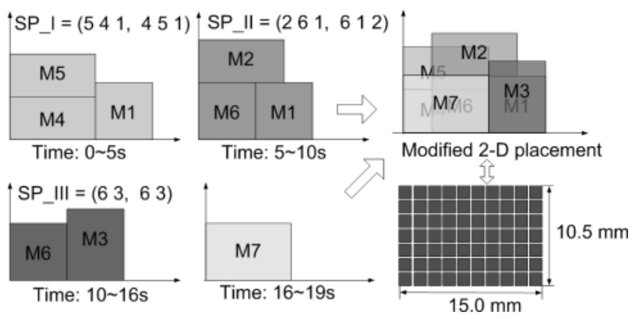


Fig. 14. Placement results obtained by the sequence pair-based method.

of sequence pairs is performed. Based on the same simulated annealing scheme as described in Section 4, we applied the SP-based method to the design of PCR biochips, whereby three kinds of generation functions were employed: (1) rotate a module; (2) interchange two modules in both sequences, that is, both Γ_+ and Γ_-; and (3) interchange two modules in only one sequence Γ_+ (or Γ_-). In addition, the algorithm based on the longest common subsequence computation (i.e., algorithm 1 in [Tang et al. [2001]) is used to evaluate the generated sequence pair. Figure 14 illustrates the result obtained using the SP-based method for the PCR example. The total area of the placement generated is 157.5 mm^2, that is, it consists of 70 cells where the pitch of each cell is 1.5 mm. The computation takes 40 seconds of CPU time. Note that we here modify the SP method to handle the dynamic reconfigurability of digital microfluidic biochips, that is, a series of 2-D configurations in different time spans should be taken together into account during placement realization from constraint graphs, as shown in Figure 14. Some microfluidic modules, for example, modules M1 and M3, can use the same cells (via dynamic reconfiguration) when their time-spans do not overlap.

Due to the efficient utilization of dynamic reconfigurability, the algorithm leads to a highly compact placement. However, the placement with the minimum array area does not provide adequate fault-tolerance. We determine the FTI of the placement shown in Figure 14 using the fast algorithm described in Section 5.3 (the calculation of the FTI takes only 1.7 seconds of CPU time). In order to facilitate the comparison, the single fault assumption is used in the evaluation examples; it can, however, be easily extended to multiple faults, as shown in Section 5.4. The input to the FTI evaluation algorithm is a modified 2-D placement, that is, a series of 2-D placements during different time spans. Thus, there are some cells contained in multiple modules, for example, the bottom-left cell is included in both M1 and M3. In order to check if it is C-Covered, we need to temporarily remove the faulty modules M1 and M3, respectively, and to check if there is sufficient empty space to accommodate the faulty module in its corresponding time-spans (e.g., $0 \sim 10$ s for M1 and $10 \sim 16$ s for M3). The bottom-left cell is deemed to be C-Covered if and only if both M1 and M2 can be successfully relocated to the fault-free region via partial reconfiguration. The FTI for this design is only 0.3429, which implies that only 24 cells in this 7×10 array are C-Covered. A microfluidics-based biochip with

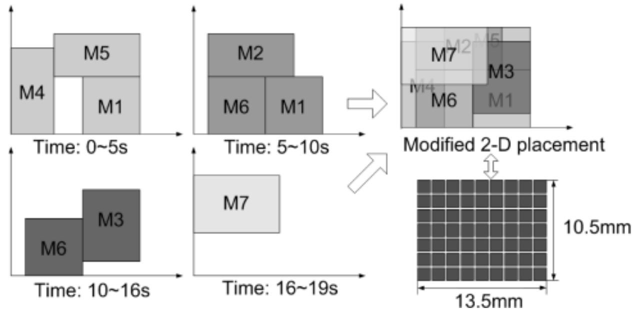


Fig. 15. Placement of Example 1 obtained from the simulated annealing-based procedure ($7 \times 9 = 63$ cells; FTI $= 0.1270$).

such a low degree of fault tolerance is not suitable for critical DNA analysis. So, we should take fault-tolerance capability into account during module placement. Therefore, the sequence pair-based method is not appropriate for such applications.

Next, we apply the proposed placement procedure described in Section 4 to this example. To compare it with the aforementioned baseline method, we first consider the minimization of the array area as the only cost metric. The placement generated by the simulated annealing procedure is shown in Figure 15. Its total area is 141.75 mm^2 (63 cells), which is 10% less compared to the baseline. The computation takes 5 minutes of CPU time. We determine its FTI value to be only 0.1270. Similar to the sequence pair baseline, this compact design has low fault-tolerance capability.

In order to increase the fault-tolerance capability for PCR, we include the FTI into the cost function for the annealing procedure. The goal of this enhanced placement algorithm is to maximize the FTI while keeping the total biochip area small. The FTI and the area are conflicting criteria because a high FTI often requires a larger biochip area. In our multiobjective placement problem, a solution is a 2-tuple (area, FTI) resulting from a feasible placement of microfluidic modules.

Weighting is a commonly used method for multiobjective optimization. A weight is assigned to each objective according to its relative importance. Next, the different objectives are combined into a single objective using a weighted sum. The solution with the lowest weighted sum is selected. In our problem, weights α and β are assigned to the criteria of area and FTI, respectively. We set α to 1 and adjusted β according to the degree of importance of fault tolerance. The solution with the lowest value of the metric ($\alpha \times$ area $-\beta \times$ *FTI*) was considered to be an acceptable solution. Based on this weighting approach, we implemented two different placement methods.

(1) *Single-Stage Simulated Annealing-Based Algorithm*. The weighted sum ($\alpha \times$ area $-\beta \times$ *FTI*) replaces the single area criterion in the cost function of Section 4. The annealing parameters are the same as in Section 4.
(2) *Two-Stage Simulated Annealing-Based Algorithm*. In the first stage, a fault-oblivious simulated annealing-based algorithm is used to obtain a

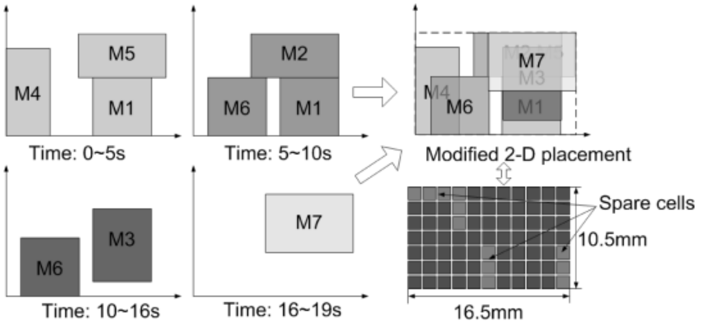


Fig. 16. Placement of Example 1 obtained from the enhanced module placement algorithm: 7 × 11 = 77 cells, and FTI = 0.8052.

placement with the smallest area. Starting from this intermediate configuration, the second stage uses low temperature simulated annealing (*LTSA*) [Chen and Chang 2005; Sechen 1988] to refine the placement, in order to enhance fault tolerance. The measure FIT is included in the cost function, while the total area is kept as small as possible. In addition, during *LTSA*, only single-module displacement is performed. Two-module interchange is not allowed since it normally causes some forbidden overlap for a compact design.

Experiments showed that the single-stage algorithm took much more computation time (687 minutes) than the two-stage method (20 minutes). This is due to the need to evaluate the FTI for each intermediate placement during the high temperature stage ($2 < T < 10000$). However, it was observed experimentally that for high temperatures, most intermediate placements belong to two categories. In the first category, these placements have forbidden overlaps and are therefore not feasible. The FTI is not relevant for these infeasible placements. In the second category, the placements are feasible but the total biochip areas are relatively large. The FTI is usually 1 in such cases. This implies that relatively less attention is paid to the FTI when the temperature is high. The annealing procedure only begins to address the FTI when the temperature is low. Thus, we note that the two-stage algorithm is able to obtain comparable results to the single-stage method, while eliminating the unnecessary computation for the FTI during high temperatures. Since the two-stage algorithm significantly outperforms the single-stage method, we use it for the fault-tolerant module placement problem.

The solution obtained by the enhanced placement algorithm requires an area of 173.25 mm^2 and yields an FTI of 0.8052; see Figure 16. In comparison to the previous placement with less area, this solution leads to an increase of 534% in the FTI while increasing the area by only 22.2%. This is clearly a more desirable placement for the safety-critical PCR assay.

6.2 Example 2: Multiplexed *In Vitro* Diagnostics

Field point-of-care clinical diagnostics is another promising application of digital microfluidics-based biochips. For example, the *in vitro* measurement of

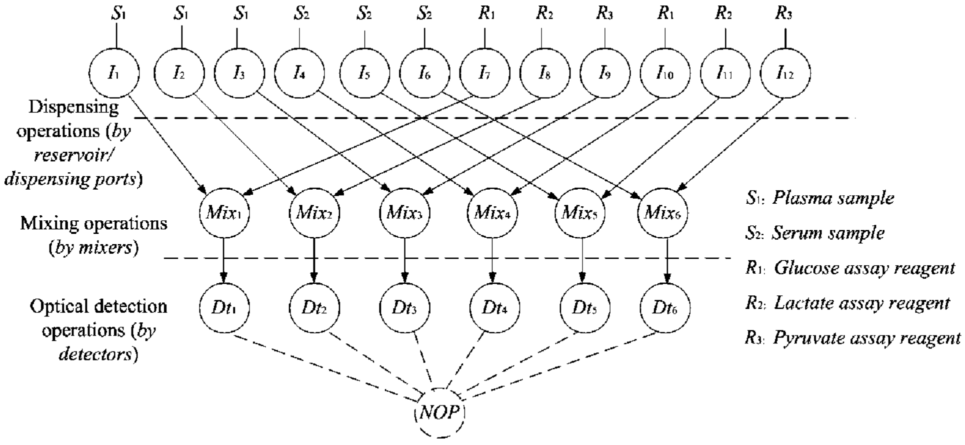


Fig. 17. Sequencing graph model for multiplexed *in vitro* diagnostics.

glucose and other metabolites (such as lactate, glutamate, and pyruvate) in human physiological fluids is of great importance in the clinical diagnosis of metabolic disorders. Recently, the feasibility of performing a colorimetric enzyme-kinetic assay (e.g., glucose assay) on a digital microfluidic biochip has been demonstrated in experiments [Srinivasan et al. 2003]. This full-custom biochip consists of a basic microfluidic platform which moves and mixes droplets containing biochemical samples and reagents, and several reservoirs that store and generate the droplets of samples and reagents. The absorbance of the assay product can be measured using an integrated optical detection system.

Using similar enzymatic reaction protocols and modified reagents, several enzyme-kinetic assays can be integrated together for multiplexed *in vitro* diagnostics on different human physiological fluids, which can be performed concurrently on a microfluidic biochip. For instance, two different types of human physiological fluidics—plasma and serum—are sampled into the microfluidic biochip, and each of them is assayed for glucose, lactate, and pyruvate measurements. The sequencing graph model of this example is shown in Figure 17; it has more nodes than the sequencing graph for the previous example. Using the integer linear programming (ILP) method, the optimal schedule for this multiplexed assay can be obtained in the architectural-level synthesis, as shown in Figure 18 [Su and Chakrabarty 2004]. Note that only the modules involved in mixing, storing, and optical detection operation are considered in our placement problem. On the other hand, locations of on-chip reservoirs/dispensing ports that implement dispensing operations can be determined manually after the placement phase, since they do not affect the area of the microfluidic array. Here, we assume that one on-chip reservoir/dispensing port is dedicated to each type of sample and reagent fluid. In addition, there is another reservoir for waste fluids. We assume that all mixing operations are carried out in 2×4-array mixers. A single cell with a segregation wrapper can be used to temporarily store the droplet. As stated earlier, these mixers and storage units can be viewed as reconfigurable virtual devices. Furthermore, we can also add

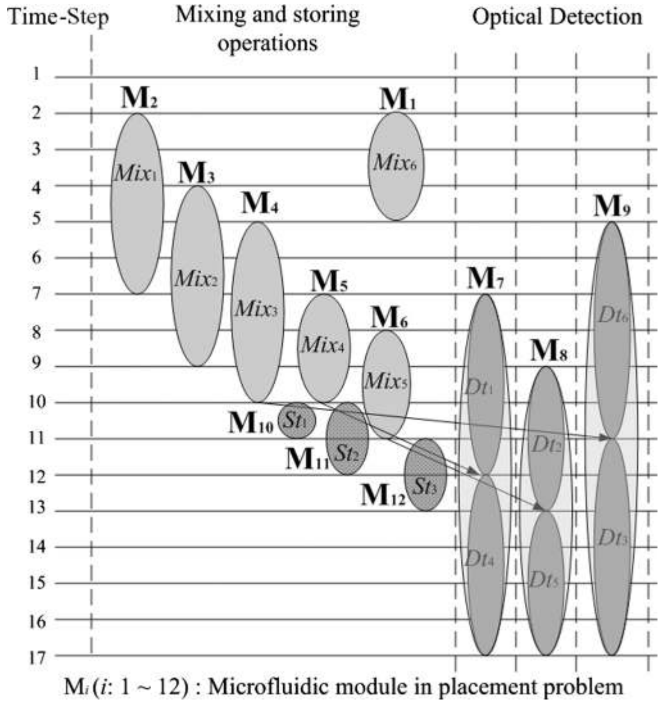


Fig. 18. Schedule highlighting the usage of microfluidic modules.

reconfigurability to the optical detection modules that represent the detection locations. Assume an array of optical detectors (e.g., CMOS optical microsensors) can be integrated to the plate of biochip. If some cell in the original optical detection module is faulty, we can easily change the detector location to a fault-free region, and then control the corresponding detector to perform the absorbance measurement. In this sense, optical detecting modules can also be considered reconfigurable.

Next, we investigate the optimal module placement for this biochip design. We first apply the proposed simulated annealing-based method to this example in order to minimize the array area. Figure 19 illustrates a highly compact placement (the result takes 15 minutes of CPU time), whereby a 10×12 microfluidic array is designed to accommodate all microfluidic modules. Reconfigurability is leveraged to allow multiple modules to share the same cells. However, we find that this compact design has poor fault-tolerance capability; its FTI is only 0.0833, which implies that most cells in this microfluidic array are not C-Covered (there are only 10 cells that are C-Covered, as highlighted in Figure 19). Consequently, if a cell becomes faulty during field operation, there is a low probability that the biochip can tolerate this fault via reconfiguration. Obviously, this situation is not desirable for the safety-critical requirement of clinical diagnostics.

We next apply the two-stage simulated annealing approach used in Section 6.1 to this example. The result generated (with 45 minutes of CPU time) is shown in Figure 20. The FTI for this placement is 0.9889, which

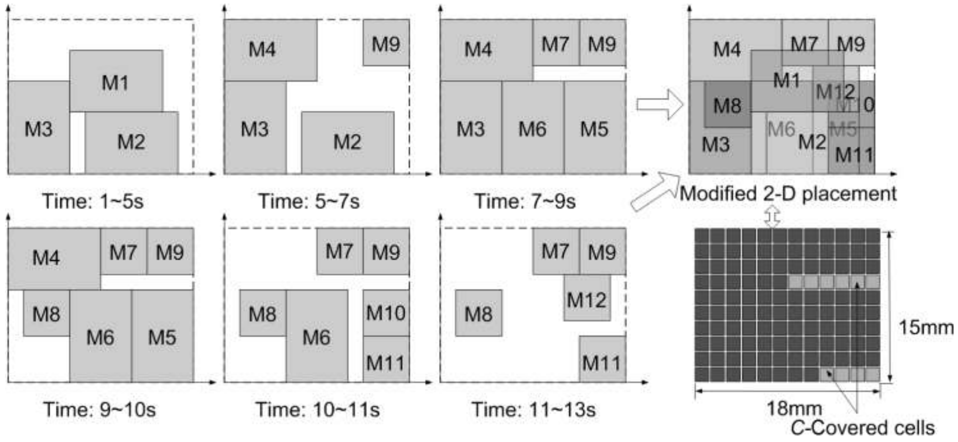


Fig. 19. Placement of Example 2 obtained from the simulated annealing-based procedure (10 × 12 = 120 cells; FTI = 0.0833).

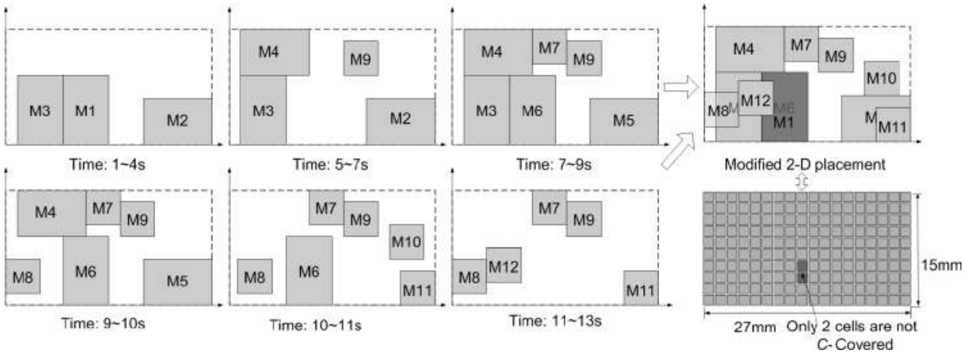


Fig. 20. Placement of Example 2 obtained from the enhanced module placement algorithm (10 × 18 = 180 cells, and FTI = 0.9889).

represents a tenfold increase compared to the previous design. As a tradeoff, the array area is increased by 50%. We illustrate the module placement and the assay operation schedule using the 3-D box model shown in Figure 21(a), where each microfluidic module is represented as a 3-D box. The projection of a 3-D box on the *x-y* plane represents the placement of this module on the microfluidic array, while the projection on the T-axis (time axis) represents the schedule of the assay operation. Note that transparent electrodes determine the locations of the integrated optical detectors active during the operation. As shown in Figure 21(b), we can further integrate optical detectors, as well as on-chip reservoirs/dispensing ports, into the microfluidic array to form a complete digital microfluidic biochip for multiplexed *in vitro* diagnostics.

6.3 Multiobjective Optimization Analysis

We further analyze the multiobjective optimization problem of module placement. The PCR example is used as an illustration.

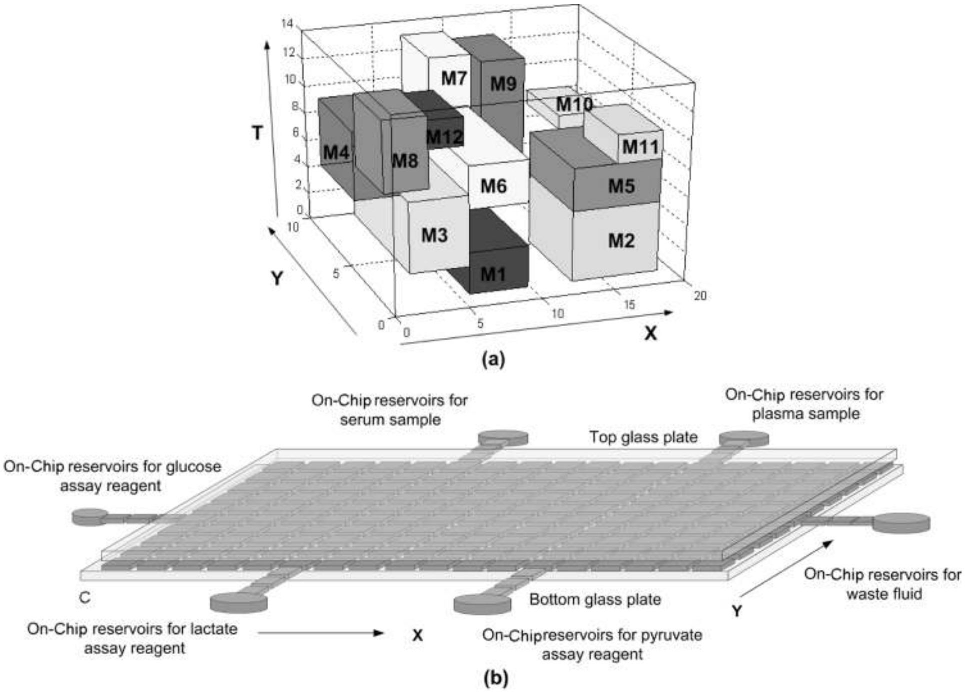

14
12
10
8
6
4
2
0
T
M7
M9
M10
M4
M8
M12
M11
M6
M5
M3
M2
M1
10
5
0
Y
0
5
10
15
20
X
(a)
On-Chip reservoirs for serum sample
On-Chip reservoirs for plasma sample
Top glass plate
On-Chip reservoirs for glucose assay reagent
On-Chip reservoirs for waste fluid
C
Bottom glass plate
Y
On-Chip reservoirs for lactate assay reagent
X
On-Chipreservoirs for pyruvate assay reagent
(b)

Fig. 21. (a) A 3-D box model to illustrate the placement result; (b) a digital microfluidic biochip design for multiplexed *in vitro* diagnostics.

Table II. Solutions for Different Values of β

β	10	20	30	40	50	60
Area (mm^2)	141.75	157.5	173.25	189.0	204.75	222.75
FTI	0.2857	0.7143	0.8052	0.8571	0.9780	1.0

6.3.1 *Effect of β.* The parameter β represents the importance of fault tolerance to the biochip designer. If fault tolerance is critical, for example, for implantable microfluidic drug-dosing systems [Reed and Lye 2004], a relatively large value of β can be used to increase the FTI. On the other hand, if fault tolerance is less important, for example, for disposable carry-home glucose detectors for one-time use [Ahn et al. 2004], a relatively small value of β can be used, thereby reducing the area and product cost.

In the final set of experiments, we varied β to investigate the relationship between chip area and the FTI (Table II). With an area of 222.75 mm^2, we can ensure that the system will always tolerate one faulty cell.

6.3.2 *Pareto Optimization.* Other than the weighting method used thus far in the article, *Pareto optimization* is an approach to deal with multiobjective optimization problems [Coello 1999]. This approach preserves the multidimensionality of the problem. Rather than selecting one solution, a set of Pareto optimal solutions is found. A solution is Pareto optimal if there exists no feasible solution that improves some criterion without causing a simultaneous deterioration in at least one other criterion. In our problem, for a given solution

Table III. Results for *MinA&FixedFTI*

FTI	0.1∼0.2	0.2∼0.3	0.3∼0.4	0.4∼0.5	0.5∼0.6	0.6∼0.7	0.7∼0.8	0.8∼0.9	0.9∼1.0
Minimized Area (mm^2)	141.75	141.75	141.75	157.5	157.5	157.5	157.5	173.25	204.75

Table IV. Results for *MaxFTI&FixedA*

Area (mm^2)	<145	145∼165	165∼185	185∼205	>205
Maximized *FTI*	0.3333	0.7143	0.8052	0.9780	1.0

(area*, FTI*), if any other solution (area, FTI) satisfies the condition that either area $\leq$ area* and FTI $<$ FTI*, or area $>$ area* and FTI $\geq$ FTI*, then this solution is Pareto optimal. A set of Pareto optimal solutions lies on the boundary of the feasible design region. This boundary, called the *Pareto front*, separates the feasible from the infeasible design region. However, it is not easy to find an explicitly analytical expression for the Pareto front that contains the Pareto optimum, since the analytical relationships between different objectives are usually unknown. Here, we develop a heuristic approach based on a modified weighting method to find the set of Pareto optimal solutions and the Pareto front for the module placement problem.

We consider the following two subproblems:

(1) *MinA&FixedFTI*. The goal here is to find a placement with the smallest biochip area such that its *FTI* value lies within a fixed range (*FTImin* $\leq$ FTI $\leq$ *FTImax*; *FTImin* and *FTImax* are known *a priori*).

(2) *MaxFTI&FixedA*. The goal here is to find a placement with the highest *FTI* value such that the array area lies within a fixed range (*Amin* $\leq$ Area $\leq$ *Amax*; *Amin* and *Amax* are known *a priori*).

The modified two-stage simulated annealing-based algorithm presented in Section 6.1 is used to solve both *MinA&FixedFTI* and *MaxFTI&FixedA*. Note that in the modified weighting method, the weight β needs to be tuned experimentally in order to find an acceptable solution. We use the PCR problem as an evaluation example. The experimental results for *MinA&FixedFTI* and *MaxFTI&FixedA* are listed in Table III and Table IV, respectively. These results can be used to estimate a Pareto front, as shown in Figure 22. The intersection points, such as (area = 141.75 mm^2, FTI = 0.3333), (area = 157.5 mm^2, FTI = 0.7143), and (area = 173.25 mm^2, FTI = 0.8052), belong to the set of Pareto optimal solutions. If we further divide the fixed range in both the *MinA&FixedFTI* and *MaxFTI&FixedA* problems, we can get more Pareto optimal solutions and make the Pareto front more accurate. The region below the Pareto front is an infeasible design region that results from the inherent tradeoff between two optimization objectives, that is, the biochip area and the FTI value.

7. CONCLUSIONS

We have presented a simulated annealing-based technique for module placement in digital microfluidics-based biochips. The placement criteria include the chip area as well as fault-tolerance; the latter allows a microfluidic module to

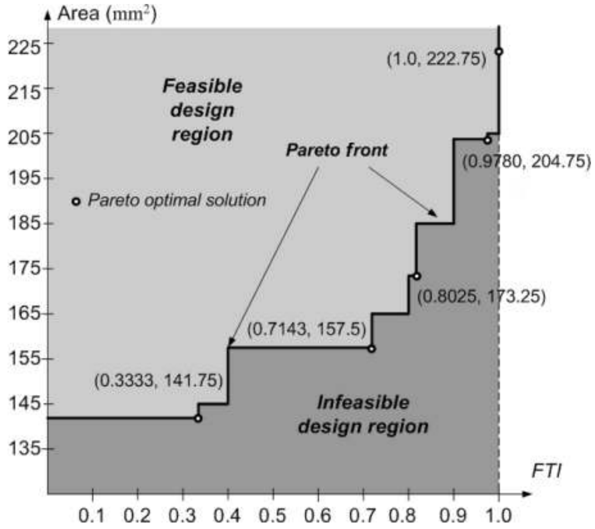


Fig. 22. Pareto optimal solutions and the Pareto front for module placement for the PCR example.

be relocated elsewhere in the system when a single cell is detected to be faulty. The placement problem accounts for the dynamic reconfigurability of droplet-based microfluidics, whereby groups of cells can be reconfigured to change their functionality during the concurrent execution of a set of bioassays. We have presented simulation results for case studies involving the polymerase chain reaction and multiplexed *in vitro* clinical diagnostics. This work is expected to facilitate the automated design of biochips, especially since their complexity is expected to grow steadily as they are increasingly used for clinical diagnosis, DNA sequencing, and other laboratory procedures involving molecular biology.

REFERENCES

AGNIHOTRI, A. R., ONO, S., LI, C., YILDIZ, M. C., KHATKHATE, A., KOH, C. K., AND MADDEN, P. H. 2005. Mixed block placement via fractional cut recursive bisection. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 24*, 748–761.

AHN, C. H., CHOI, J.-W., BEAUCAGE, G., NEVIN, J. H., LEE, J.-B., PUNTAMBEKAR, A., AND LEE, J. Y. 2004. Disposable smart lab on a chip for point-of-care clinical diagnostics. In *Proceedings of the IEEE 92*, 154–173.

ANDRONESCU, M., D., DEES, D., SLAYBAUGH, L., ZHAO, Y., CONDON, A., COHEN, C., AND SKIENA, S. 2003. Algorithms for testing that sets of DNA words concatenate without secondary structure. *Natural Comput. 2*, 391–415.

BAZARGAN, K., KASTNER, R., AND SARRAFZADEH, M. 2000a. Fast template placement for reconfigurable computing systems. *IEEE Des. Test Comput. 17*, 68–83.

BAZARGAN, K., KASTNER, R., AND SARRAFZADEH, M. 2000b. 3-D floorplanning: Simulated annealing and greedy placement methods for reconfigurable computing systems. In *Design Automation for Embedded Systems 5*, 329–338.

BÖHRINGER, K. F. 2005. Modeling and controlling parallel tasks in droplet-based microfluidic systems. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst.*

BRADLEY, R. AND SKIENA, S. 1997. Fabricated arrays of strings. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, 57–66.

CASOTTO, A., ROMEO, F., AND SANGIOVANNI-VINCENTELLI, A. 1987. A parallel simulated annealing algorithm for the placement of macro-cells. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 6*, 838–847.

CHANDY, J. A., KIM, S., RAMKUMAR, B., PARKES, S., AND BANERJEE, P. 1997. An evaluation of parallel simulated annealing strategies with application to standard cell placement. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 16*, 398–410.

CHANG, Y. C., CHANG, Y. W., WU, G. M., AND WU, S. W. 2000. B*-tree: A new representation for nonslicing floorplans. In *Proceedings of the IEEE/ACM Design Automation Conference*, 458–463.

CHATTERJEE, A. N. AND ALURU, N. R. 2005. Combined circuit/device modeling and simulation of integrated microfluidic systems. *J. Microelectromechanical Syst. 14*, 81–95.

CHEN, T.-C. AND CHANG, Y.-W. 2005. Modern floorplanning based on fast simulated annealing. In *Proceedings of the ACM International Symposium on Physical Design (ISPD)*, 104–112.

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

COELLO, C. 1999. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In *Proceedings of the IEEE Evolutionary Computation Conference*, 140–143.

CONG, J., HUANG, H., AND YUAN, X. 2005. Technology mapping and architecture evaluation for k/m-macrocell-based FPGAs. *ACM Trans. Des. Autom. Electro. Syst. 10*, 3–23.

CONG, J. AND LIM, S. 2004. Retiming-Based timing analysis with an application to mincut-based global placement. *IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 23*, 1684–1692.

DING, J., CHAKRABARTY, K., AND FAIR, R. B. 2001. Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 20*, 1463–1468.

EDMONDS, J., GRYZ, J., LIANG, D., AND MILLER, R. J. 2003. Mining for empty spaces in large data sets. *Theoretical Computer Science 296*, 435–452.

GAREY M. AND JOHNSON, D. 1979. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, New York.

GRIFFITH, E. AND AKELLA, S. 2005. Performance characterization of a reconfigurable planar array digital microfluidic system. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst*.

GUO, P.-N., CHENG, C.-K., AND YOSHIMURA, T. 1999. An O-tree representation of nonslicing floorplan and its applications. In *Proceedings of the ACM/IEEE Design Automation Conference*, 268–273.

HANDA, M. AND VEMURI, R. 2004. A fast algorithm for finding maximal-empty rectangles. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 744–745.

INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTOR (ITRS). http://public.itrs.net/Files/2003ITRS/Home2003.htm.

JONES, T. B., GUNJI, M., WASHIZU, M., AND FELDMAN, M. J. 2001. Dielectrophoretic liquid actuation and nanodroplet formation. *J. Appl. Phys. 89*, 1441–1448.

KAHNG, A. B., MANDOIU, I. I., REDA, S., XU, X., AND ZELIKOVSKY, A. Z. 2005. Computer-Aided optimization of DNA array design and manufacturing. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst*.

LIN, J. M. AND CHANG, Y. W. 2001. TCG: A transitive closure graph-based representation for nonslicing floorplans. In *Proceedings of the IEEE/ACM Design Automation Conference*, 764–769.

MCCLUSKEY, P. 2002. Design for reliability of micro-electro-mechanical systems (MEMS). In *Proceedings of the IEEE Electronic Components and Technology Conference*, 760–762.

MELTZER, S. 1998. *PCR in Bioanalysis*. Humana Press, Totowa, N.J.

MUKHERJEE, T. AND FEDDER, G. K. 1998. Design methodology for mixed-domain systems-on-a-chip [MEMS design]. In *Proceedings of the IEEE VLSI System Level Design Conference*, 96–101.

MURATA, H., FUJIYOSHI, K., NAKATAKE, S., AND KAJITANI, Y. 1995. Rectangle-packing-based module placement. In *Proceedings of the IEEE International Conference on CAD*, 472–479.

PAIK, P., PAMULA, V. K., AND FAIR, R. B. 2003. Rapid droplet mixers for digital microfluidic systems. *Lab on a Chip 3*, 253–259.

PFEIFFER, A. J., MUKHERJEE T., AND HAUAN, S. 2005. Synthesis of multiplexed biofluidic microchips. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst.*.

POLLACK, M. G. 2001. *Electrowetting-Based Microactuation of Droplets for Digital Microfluidics*. PhD thesis, Duke University.

POLLACK, M. G., SHENDEROV, A. D., AND FAIR, R. B. 2002. Electrowetting-Based actuation of droplets for integrated microfluidics. *Lab on a Chip 2*, 96–101.

REED, M. L. AND LYE, W. K. 2004. Microsystems for drug and gene delivery. In *Proceedings of the IEEE 92*, 56–75.

SAIT, S. AND YOUSSEF, H. 1995. *VLSI Physical Design Automation: Theory and Practice*, IEEE Press, New York.

SAKANUSHI, K. AND KAJITANI, Y. 2000. The quarter-state sequence (Q-sequence) to represent the floorplan and applications to layout optimization. In *Proceedings of the Asia Pacific Conference Circuits and Systems* 829–832.

SARRAFZADEH, M. AND WONG, C. K. 1996. *An Introduction to VLSI Physical Design*. McGraw-Hill, New York.

SECHEN C. 1988. *VLSI Placement and Global Routing Using Simulated Annealing*. Kluwer Academic, Boston, Mass.

SECHEN,C. AND SANGIOVANNI-VINCENTELLI, A. 1985. The TimberWolf placement and routing package. *IEEE J. Solid-State Circuits 20*, 510–522.

SHAPIRO, B., MOON, H., GARRELL, R. AND KIM, C. J. 2003. Modeling of electrowetted surface tension for addressable microfluidic systems: Dominant physical effects, material dependences, and limiting phenomena. In *Proceedings of the IEEE Conference MEMS*, 201– 205.

SRINIVASAN, V., PAMULA, V. K., POLLACK, M. G., AND FAIR, R. B. 2003. Clinical diagnostics on human whole blood, plasma, serum, urine, saliva, sweat, and tears on a digital microfluidic platform. In *Proceedings of the μTAS*, 1287–1290.

SU, F. AND CHAKRABARTY, K. 2004. Architectural-level synthesis of digital microfluidics-based biochips. In *Proceedings of the IEEE International Conference on CAD*, 223–228.

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

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

TANG, X., TIAN, R., AND WONG, D. F. 2001. Fast evaluation of sequence pair in block placement by longest common subsequence computation. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 20*, 1406–1413.

TANG, X. T. AND WONG, D. F. 2001. FAST-SP: A fast algorithm for block based on sequence pair. In *Proceedings of the Asia and South Pacific Design Automation Conference*, 521–526.

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

WANG, X., WHITE, J., KANAPKA, J., YE, W., AND ALURU, N. 2005. Algorithms in FastStokes and its application to micromachined device simulation. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst*.

WHITE, J. 2004. CAD challenges in BioMEMS design. In *Proceedings of the IEEE/ACM Design Automation Conference*, 629–632.

YANG, P. AND CHERN, J. 1993. Design for reliability: The major challenge for VLSI. In *Proceedings of the IEEE 58*, 730–744.

YOUNG, E. F. Y., CHU, C. C. N., AND SHEN, Z. C. 2003. Twin binary sequences: A nonredundant representation for general nonslicing floorplan. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 22*, 457–469.

YUH, P. H., YANG, C. L., AND CHANG, Y. W. 2004a. Temporal floorplanning using the T-tree formulation. In *Proceedings of the IEEE International Conference on CAD* , 300–305.

YUH, P. H., YANG, C. L., CHANG, Y. W., AND CHEN, H. L. 2004b. Temporal floorplanning using 3D-subTCG. In *Proceedings of the Asia and South Pacific Design Automation Conference*, 725–730.

ZENG, J. AND KORSMEYER, F. T. 2004. Principles of droplet electrohydrodynamics for lab-on-a-chip. *Lab on a Chip 4*, 265–277.

ZHANG, T., CHAKRABARTY, K., AND FAIR, R. B. 2002. *Microelectrofluidic Systems: Modeling and Simulation*. CRC Press, Boca Raton, Fla.

Received August 2005; revised February 2006; accepted February 2006