Unified High-Level Synthesis and Module Placement for Defect-Tolerant Microfluidic Biochips*

Fei Su and Krishnendu Chakrabarty
Department of Electrical and Computer Engineering
Duke University, Durham, NC 27708, USA
{fs, krish}@ee.duke.edu

ABSTRACT

Microfluidic biochips promise to revolutionize biosensing and clinical diagnostics. As more bioassays are executed concurrently on a biochip, system integration and design complexity are expected to increase dramatically. This problem is also identified by the 2003 ITRS document as a major system-level design challenge beyond 2009. We focus here on the automated design of droplet-based microfluidic biochips. We present a synthesis methodology that unifies operation scheduling, resource binding, and module placement for such "digital" biochips. The proposed technique, which is based on parallel recombinative simulated annealing, can also be used after fabrication to bypass defective cells in the microfluidic array. A real-life protein assay is used to evaluate the synthesis methodology.

Categories and Subject Descriptors

B.7.2 B.7.3 [**Integrated Circuits**]: Design Aids, Reliability and Testing.

General Terms

Algorithms, Performance, Design, Reliability

Keywords

Synthesis, placement, defect tolerance, microfluidics, biochip.

1. INTRODUCTION

Recent advances in microfluidics technology have generated tremendous interest in the design and implementation of miniaturized devices for biomolecular recognition. These devices, referred to interchangeably in the literature as microfluidic biochips, lab-on-a-chip and bioMEMS, promise to revolutionize biosensing, clinical diagnostics and drug discovery due to their small size, use of nanoliter sample volumes, lower cost, and higher sensitivity compared to conventional laboratory methods [1].

The first generation of microfluidic biochips, based on the principle of continuous fluid flow, consists of microchannels, micropumps, and microvalves permanently etched in silicon or glass substrates. In contrast, the second generation of microfluidic biochips is based on the manipulation of liquids as discrete nano-liter droplets. Following the analogy of digital electronics, this technology is referred to as "digital microfluidics" [1].

As more bioassays are executed concurrently on a digital microfluidic biochip, system integration and design complexity are expected to increase dramatically. The 2003 International Technology Roadmap for Semiconductors (ITRS) clearly identifies the integration of electrochemical and electro-biological techniques as one of the system-level design challenges that will be faced beyond 2009, when feature sizes shrink below 50 nm [2]. As in the case of integrated circuits, increase in the density and area of microfluidic biochips may reduce yield, especially for smaller feature sizes. It will take time to ramp up the yield based on an understanding of defects in such biochips. Therefore, defect tolerance for microfluidic biochips is especially important for the emerging marketplace. Unfortunately, current full-custom biochip design techniques may not scale well for large designs, and they do not easily handle defect tolerance issues. Thus, there is a pressing need to deliver the same level of CAD support to the biochip designer that the semiconductor industry now takes for granted.

Most design automation research for digital microfluidic biochips has been focused on device-level physical modeling and simulation of single components [3]. While top-down system-level design tools are now commonplace in IC design, few such efforts have been reported for microfluidic biochips. These synthesis tools can relieve biochip users from the burden of manually optimizing a set of assays for increased throughout. 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 the development of the nano- and micro-scale bioassays, leaving implementation details to the synthesis tools.

The synthesis of a digital microfluidic biochip can be divided into two major phases, broadly referred to as high-level synthesis and physical design [4]. High-level synthesis is used to generate a macroscopic structure of the biochip from the behavioral model for a bioassay; this structure is analogous to a structural RTL model in electronic CAD. Physical design creates the final layout of the biochip, consisting of the placement of microfluidic modules such as mixers and storage units, as well as the routes that droplets take between different modules, locations of on-chip reservoirs, dispensing ports and integrated optical detectors, and other geometrical details.

A recent approach for biochip synthesis decouples high-level synthesis from physical design [4]. It is based on rough estimates for placement costs such as the areas of the microfluidic modules. These estimates provide lower bounds on the exact biochip area, since the overheads due to spare cells and cells used for droplet transportation are not known *a priori*. It cannot be accurately predicted if the biochip design meets system specifications, e.g., maximum allowable array area, until both high-level synthesis and physical design are carried out. When design specifications are not met, time-consuming iterations between high-level synthesis and physical design are required. A link between these steps is especially necessary if defect tolerance is to be considered during synthesis.

* This research was supported by the National Science Foundation under grant number IIS-0312352.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

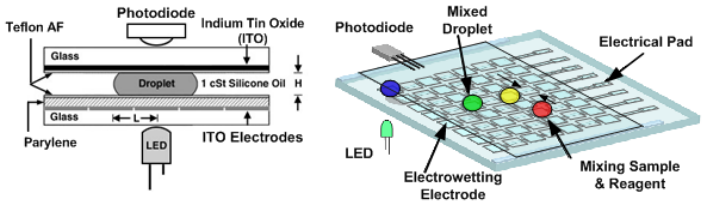

Figure 1. Schematic of a digital microfluidic biochip.

In this paper, we propose a synthesis methodology that unifies operation scheduling, resource binding, and module placement. The proposed algorithm is based on parallel recombinative simulated annealing (PRSA). All three tasks, i.e., resource binding, scheduling, and placement, are carried out at each step of the algorithm. Thus, exact placement information, instead of a crude area estimate, is used to judge the quality of architectural-level synthesis. This information is utilized by the annealing process to select resources and schedule bioassay operations to produce a high-quality design. This method allows architectural design and physical design decisions to be made simultaneously. Moreover, defect tolerance can be easily incorporated during synthesis, whereby resources for bioassay functions are carefully selected and placed in the array to bypass defective cells; in this way, the bioassay functionality is not compromised. We use a large-scale protein assay to evaluate the proposed synthesis methodology.

The result of the paper is organized as follows. In Section 2, related prior work is discussed. Next we introduce an overview of digital microfluidic biochips in Section 3. Section 4 presents the PRSA-based synthesis method. An enhancement to the algorithm for defect tolerance is also discussed. In Section 5, we use a protein assay as a case study to evaluate the proposed synthesis methodology. Finally, conclusions are drawn in Section 6.

2. RELATED PRIOR WORK

Synthesis of integrated circuits is a well-studied problem [5]. Driven by deep submicron considerations, the incorporation of physical design issues in high-level synthesis has received considerable attention recently [6, 7]. Defect tolerance for integrated circuits has also emerged as an important design criterion for yield enhancement [8].

MEMS design is a relatively young field compared to IC design. Nevertheless, a number of MEMS CAD tools are available today for modeling [9] and synthesis [10]. Attempts have also been made to make MEMS defect-tolerant [11]. However, because of the differences in actuation methods between MEMS and digital microfluidics, these techniques cannot be directly used for the design of microfluidic biochips.

Although research on microfluidic biochips has gained momentum in recent years, CAD tools for biochips are still in their infancy [12]. Some commercial computational fluid dynamics (CFD) tools, such as CFD-ACE+ from CFD Research Corporation, support 3D simulation of microfluidic transport. A recent release of CoventorWare from Coventor, Inc. includes microfluidic behavioral models to allow top-down system-level design for continuous-flow systems. Recently, classical architectural-level synthesis techniques, oblivious of physical design issues, have been applied to digital microfluidic biochips [4].

3. DIGITAL MICROFLUIDIC BIOCHIPS

The microfluidic biochips discussed in this paper are based on the manipulation of nanoliter droplets using the principle of electrowetting, i.e., modulation of the interfacial tension between a conductive fluid and a solid electrode through an electric field. The basic cell of a digital microfluidic biochip consists of two parallel

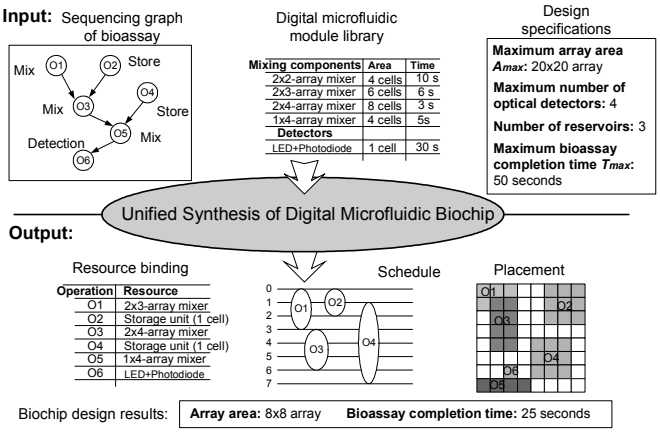

Figure 2. An example illustrating system-level synthesis.

glass plates, as shown in Figure 1. The bottom plate contains a patterned array of individually controllable electrodes, and the top plate is coated with a continuous ground electrode. The droplets containing biochemical samples and the filler medium, such as the silicone oil, are sandwiched between the plates. The droplets travel inside the filler medium. By varying the electrical potential along a linear array of electrodes, droplets can be moved along this line of electrodes. The velocity of the droplet can be controlled by adjusting the control voltage (0~90V), and droplets can be moved at speeds of up to 20 cm/s [1]. Based on this principle, microfluidic droplets can be moved freely to any location of a two-dimensional array without the need for micropumps and microvalves.

Using a two-dimensional array, many basic microfluidic operations for different bioassays can be performed, such as sample introduction (*dispense*), sample movement (*transport*), temporarily sample preservation (*store*), sample dilution with buffer (*dilute*) and the mixing of different samples (*mix)*. For instance, the *mix* operation is used to route two droplets to the same location and then turn them around some pivot points. Note that these operations can be performed anywhere on the array, whereas in continuous-flow biochips they must operate in a specific micromixer or microchamber fixed on a substrate. This property is referred to as the reconfigurability of a digital microfluidic biochip. The configurations of the microfluidic array, i.e., the routes that droplets transport and the rendezvous points of droplets, are programmed into a microcontroller that controls the voltages of electrodes in the array. In this sense, these mixers and storage units used during the operations can be viewed as reconfigurable virtual devices. In addition, a digital microfluidic biochip also contains on-chip reservoirs/dispensing ports that are used to generate and dispense sample or reagent droplets, as well as integrated optical detectors such as LEDs and photodiodes. In contrast to mixers or storage units, these resources are non-reconfigurable.

4. UNIFIED SYNTHESIS METHODOLOGY

4.1 Problem Formulation

Figure 2 illustrates the design flow for the proposed synthesis methodology. A sequencing graph is first obtained from the protocol for a bioassay [4]. This acyclic graph $G(V, E)$ has vertex set $V = \{v_i: i = 0, 1,\ldots, k\}$ in one-to-one correspondence with the set of assay operations, and edge set $E = \{(v_i, v_j): i, j = 0, 1,\ldots, k\}$ representing dependencies between assay operations. The weight for each node, $d(v_i)$, denotes the time taken for operation v_i; note however that this value is not assigned until resource binding has been performed during synthesis. Since droplet movement is very fast in contrast to assay operations [1, 13], we can ignore the

droplet transportation time between different assay operations. In addition, a microfluidic module library is also provided as an input of the synthesis procedure. This module library, analogous to a 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. Moreover, some design specifications are also given *a priori*, e.g., an upper limit on the completion time T_{max}, an upper limit on the size of microfluidic array A_{max}, and the set of non-reconfigurable resources such as on-chip reservoirs/dispensing ports and integrated optical detectors.

The proposed synthesis tool performs scheduling, resource binding, and placement in a unified manner. As in the case of high-level synthesis for ICs, resource binding refers to the mapping from bioassay operations to available functional resources. Note that there may be several types of resources for any given bioassay operation. For example, a 2×2-array mixer, a 2×3-array mixer and a 2×4-array mixer can be used for a droplet mixing operation [1]. These mixers differ in their areas as well as mixing times. In such cases, a resource selection procedure must be used. On the other hand, due to the resource constraints, a resource binding may associate one functional resource with several assay operations; this necessitates resource sharing. Once resource binding is carried out, the time duration for each bioassay operation can be easily determined. Scheduling determines the start times and stop times of all assay operations, subject to the precedence constraints imposed by the sequencing graph. In a valid schedule, assay operations that share a microfluidic module cannot execute concurrently. Scheduling and resource binding also need to be tied to the placement problem for biochips; placement determines the various configurations of a microfluidic array as well as the locations of integrated optical detectors and reservoirs/ dispensing ports. The property of virtual devices makes the placement of reconfigurable microfluidic modules, such as mixers or storage units, on a 2-D microfluidic array quite different from the traditional placement problem in VLSI design.

The output of the synthesis tool includes the mapping of assay operation to resources, a schedule for the assay operations, and the placement of the modules. The synthesis procedure attempts to find a desirable design point that satisfies the input specifications. If such a solution does not exist, the synthesis tool outputs the best solution that can be achieved. In order to measure the quality of a synthesis flow, we need to consider the minimization of the array area A and the completion time T for the bioassay. For this multi-objective optimization problem, a weighting approach is used. Here weights α and $(1-\alpha)$, where $0 < \alpha < 1$, are assigned to the criteria of normalized area (denoted by A/A_{max}) and normalized bioassay time (denoted by T/T_{max}), respectively. The solution with the lowest value of the metric $(\alpha \times A/A_{max} + (1-\alpha) \times T/T_{max})$ is considered to be an acceptable solution.

4.2 PRSA-Based Algorithm

The resource-constrained scheduling problem and the module placement problem have been shown in the literature to be *NP*-complete [14]. Therefore, heuristics are needed to solve the optimization problem in a computationally efficient manner. Parallel recombinative simulated annealing (PRSA) is a well-studied combinational optimization method that has some of the best attributes of both genetic algorithms and simulated annealing algorithms [15]. This class of algorithms is best viewed as genetic algorithms that use *Boltzmann trials* between modified and existing solutions to select the solutions that exist in the next generation.

```
PRSA-based algorithm
Set initial population of chromosome and the initial temperature T∞;
Implement the synthesis using the information of initial chromosomes:
  {Phase I: Resource binding; Phase II: Scheduling; Phase III: Placement}
while (Stopping criteria of annealing is not satisfied)
  for i = 1: N  /* Inner loop of annealing process */
    Find fitness values of chromosomes through construction procedure;
    Reproduction; /* Best chromosomes copied to the next generation */
    Crossover: {Parameterized uniform crossover is to generate the child
        chromosome from two randomly-selected parent chromosomes}
      if Fitness(child) < Fitness(parents) or
          rand(0, 1) < exp(-[Fitness(child)−Fitness(parents)]/T)
        Child chromosome is selected;
      else Parent chromosome (the best one) is selected;
      end if  /* Here a Boltzmann trial is performed */
    Mutation;  /* New chromosomes are generated randomly */
    New population replaces the old generation; end for
  T= rate × T;  /* update the temperature */ end while
Find the best chromosome from the final population;
Output the results of resource binding, scheduling and placement.
```

Figure 3. Pseudo-code for the PRSA-based heuristic algorithm.

We present a PRSA-based algorithm to solve the optimization problem for biochip synthesis. The pseudocode for this heuristic approach is shown in Figure 3. Some details of the procedure are listed as follows, and they also will be illustrated in Section 5.

4.2.1 Representation of a chromosome

A robust representation technique called *random keys* is used in this algorithm [16]. A random key is a random number sampled from [0, 1]. Each chromosome in the population can be encoded as a vector of random keys, named *genes*. *Chromosome* = {*gene*(1), ..., *gene*(*k*), *gene*(*k*+1),..., *gene*(2*k*), *gene*(2*k*+1),..., *gene*(3*k*)}, where k is the number of assay operations. Here the first set of k genes are used to determine resource binding, i.e., $Rb(i) = gene(i)$, $i = 1$ to k. The second set of k genes are to set the delay time of the operations, which is calculated as follows: delay value of operation $Dv(i) = d \times gene(i+k)$, $i = 1$ *to* k, where d is a constant that can be fine-tuned through experiments. The last k genes are used to determine the placement priorities, i.e., priority value of operation $Pv(i) = gene(i+2k)$, $i = 1$ to k.

4.2.2 Construction procedure

The goal of this procedure is to carry out resource binding, scheduling and placement under dependency and resource constraints, by using a vector of random numbers (i.e., genes from a chromosome). It consists of the following three phases.

a) Phase I: *Resource binding*

To simplify the synthesis procedure, in this phase we temporarily do not consider an upper limit on the number of available reconfigurable resources. A reconfigurable resource type for a bioassay is selected based on its associated gene value, i.e., $Rb(i)$. For example, for a mixing operation v_i, a 2×2-array mixer is selected if $Rb(i) < 0.25$; a 2×3-array mixer is chosen if $0.25 \le Rb(i) < 0.5$; a 2×4-array mixer is selected if $0.5 \le Rb(i) < 0.75$; a 4-electrode linear array mixer is selected if $Rb(i) \ge 0.75$.

Reservoirs/dispensing ports and optical detectors are non-reconfigurable resources. The number of such resources is fixed, and it is determined by the system design specifications. The gene values for the corresponding operations determine the selection of resource instance. For example, if there are two optical detectors available, namely OD_1 and OD_2, a optical detection operation v_i is bound to OD_1 if $Rb(i) < 0.5$, and to OD_2 if $Rb(i) \ge 0.5$.

After Phase I, a weight $d(v_i)$, i.e., the duration time for the corresponding operation, has been assigned to each node v_i of the

sequencing graph. Thus, an original sequencing graph without node weights is modified to a weighted sequencing graph.

b) Phase II: *Scheduling*

In this phase, a feasible bioassay schedule, satisfying temporal precedence constraints as well as non-reconfigurable resource constraints, is constructed by using the delay values $Dv(i)$ from a chromosome. Due to its low computational complexity of $O(n)$, where n is the number of operations to schedule, a *list scheduling* algorithm is used in this step [5]. As in Phase I, only constraints for non-reconfigurable resources are taken into account here.

To schedule the operation i, we set its start time to be $Start(i) = \max\{Stop(j)$: either j is the predecessor of i or it used the same resource as $i\} + Dv(i)$, and stop time as $Stop(i) = Start(i) + d(v_i)$. After this phase, a scheduled sequencing graph with resource binding is obtained.

c) Phase III: *Placement*

Based on the results from resource binding and scheduling, we attempt to place the microfluidic modules on a 2-D array to satisfy the design specifications. A greedy algorithm is used in this phase. Microfluidic modules are first sorted in the descending order of their priority values $Pv(i)$. In each step, the module with the highest priority among the unplaced ones is selected and placed. We determine an available location for this module while attempting to minimize the array area. Resource constraints must be satisfied, e.g., there should be no spatial overlap between the module with previously-placed ones if their usage overlaps in the schedule. The placement problem can also be modeled by a 3-D packing problem, which will be illustrated by an example in Section 5. In addition, we add a segregation region between two active modules. This additional area not only isolates the functional module from its neighbors, thereby avoiding unexpected cross-contamination, but it also provides a transportation path for droplet movement between different modules.

The above greedy algorithm not only deals with the placement of reconfigurable resources, but it can also determine the location of non-reconfigurable resources such as optical detectors. On the other hand, the locations of reservoirs/dispensing ports can be determined manually after synthesis, since they do not affect the area of microfluidic array or the processing time for the bioassay.

Therefore, based on the information provided by a chromosome, the synthesis procedure can be carried out based on the above three phases. The fitness value of this chromosome is determined by the synthesis results. Through a series of generations of evolution controlled by a simulated annealing process, we can find a best chromosome, i.e., with the smallest fitness value, from the final population. The synthesis results obtained from this chromosome represent the solution to our optimization problem.

4.3 Enhancement for Defect Tolerance

After the synthesis procedure is executed, digital microfluidic biochips are fabricated using standard microfabrication techniques [1]. Due to the underlying mixed technology and multiple energy domains, they exhibit unique failure mechanisms and defects. A manufactured microfluidic array may contain several defective cells. We have observed defects such as dielectric breakdown, shorts between adjacent electrodes, and electrode degradation.

Reconfiguration techniques can be used to bypass faulty cells or faulty optical detectors to tolerate manufacturing defects. Bioassay operations bound to these faulty resources in the original design need to be remapped to other fault-free resources. Due to the strict resource constraints in the fabricated biochip, alterations in the resource binding operation, schedule and placement must be carried out carefully. Our proposed system-level synthesis tool can

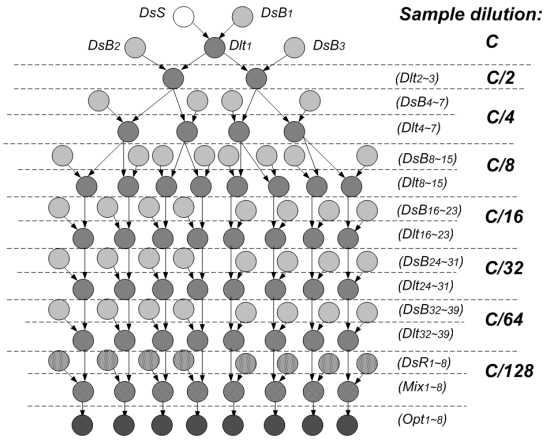

Figure 4. Sequencing graph for a protein assay.

be easily modified to deal with this issue. To reconfigure a defective biochip, a PRSA-based algorithm along the lines of the one described in Section 4.2 is used. The following additional considerations must be taken into account.

a) The objective during reconfiguration is to minimize the bioassay completion time while accommodating all microfluidic modules and optical detectors in the fabricated microfluidic array.

b) As resource constraints, the defect-free parts of the microfluidic array and the number of fabricated fault-free non-reconfigurable resources replace the original design specifications.

c) In the placement phase, the locations of the defective cells are no longer available. Note that the locations of non-reconfigurable resources such as integrated optical detectors and reservoirs/ dispensing ports are fixed in the fabricated biochip.

Using this enhanced synthesis tool, a set of bioassays can be easily mapped to a biochip with a few defective cells; thus we do not need to discard the defective biochip.

5. EXPERIMENTAL EVALUATION

We evaluate the proposed synthesis method by using it to design a biochip for a real-life protein assay. Recently, the feasibility of performing a colorimetric protein assay on a digital microfluidic biochip has been successfully demonstrated [13]. Based on the Bradford reaction [13], the protocol for a generic droplet-based colorimetric protein assay is as follows. First, a droplet of the sample, such as serum or some other physiological fluid containing protein, is generated and dispensed into the biochip. Buffer droplets, such as 1M NaOH solution, are then introduced to dilute the sample to obtain a desired dilution factor (DF). This on-chip dilution is performed using multiple hierarchies of binary mixing/splitting phases, referred to as the interpolating serial dilution method [1]. The mixing of a sample droplet of protein concentration C and a unit buffer droplet results in a droplet with twice the unit volume, and concentration $C/2$. Splitting this large droplet results in two unit-volume droplets of concentration $C/2$ each. Continuing this step in a recursive manner using diluted droplets as samples, an exponential dilution factor of $DF = 2^N$ can be obtained in N steps. After dilution, droplets of reagents, such as Coomassie brilliant blue G-250 dye, are dispensed into the chip, and they mix with the diluted sample droplets. Next the mixed droplet is transported to a transparent electrode, where an optical detector (e.g., a LED-photodiode setup) is integrated. The protein concentration can be measured from the absorbance of the products of this colorimetric reaction using a rate kinetic method [13]. Finally, after the assay is completed, all droplets are transported from the array to the waste reservoir.

A sequencing graph model can be developed from the above protocol for a protein assay (DF = 128), as shown in Figure 4. There are a total of 103 nodes in one-to-one correspondence with the set of operations in a protein assay, where DsS, DsB_i (i = 1, …, 39), and DsR_i (i = 1, …, 8) represents the generation and dispensing of sample, buffer and reagent droplets, respectively. In addition, Dlt_i (i = 1, …, 39) denotes the binary dilution (including mixing/splitting) operations, Mix_i (i = 1, …, 8) represents the mixing of diluted sample droplets, and reagent droplets; Opt_i (i = 1, …, 8) denotes the optical detection of the mixed droplets. Until the fourth step of a serial dilution, all diluted sample droplets are retained in the microfluidic array. After that stage, for each binary dilution step, only one diluted sample droplet is retained after splitting, while the other droplet is moved to the waste reservoir.

The basic operations for protein assay have been implemented on a digital microfluidic biochip [1, 13]. Experiments indicate that the dispensing operation takes 7 seconds when we use a reservoir of 4 mm diameter and a dispensing channel comprising 750 μm pitch electrodes with 100 μm channel gap [1]. Mixing is an important, yet difficult, microfluidic operation. Linear array mixing and 2-D array mixing have been performed on a biochip, and the operation times of various mixers have been found to be different [1]. Note that in these experiments, cells in mixers were assumed to have the same geometric parameters, i.e., a 1.5 mm electrode size and the 600 μm gap between the two plates. A binary dilution operation can also be easily implemented by using a linear array or 2-D array, whereby the mixing of sample droplet and buffer droplet is followed by droplet splitting. Absorbance of the assay product can be measured using an integrated LED-photodiode setup. Experiments indicate this absorbance measurement takes 30 seconds [13]. Thus we can build a microfluidic module library for a protein assay, as shown in Table 1.

We also need to specify some design parameters for the biochip to be synthesized. As an example, we set the maximum microfluidic array size to be 10×10 cells, and the maximum allowable completion time for the protein assay to be 400 seconds. We assume that there is only one on-chip reservoirs/dispensing port available for sample fluids, but two such ports for buffer fluids, two for reagent fluids, and one for waste fluids. Finally, we assume that at most four optical detectors can be integrated into this biochip.

Before applying the proposed synthesis method to the above problem instance, we use two baseline techniques to design the biochips for protein assays. In the first design, we attempt to minimize the microfluidic array size as much as possible, as shown in Figure 5(a). Only one linear array, i.e., a 4-electrode linear array, is used as both the mixer and the dilutor. It also provides the location for optical detection. Moreover, three additional storage units are needed to store the intermediate droplets, i.e., diluted samples. Due to the high resource constraint in this design, the

Table 1: Module library for synthesis.

Operation	Resource	Time (s)
DsS; *DsB*; *DsR*	On-chip reservoir/dispensing port	7
Dlt	2x2-array dilutor	12
	2x3-array dilutor	8
	2x4-array dilutor	5
	4-electrode linear array dilutor	7
Mix	2x2-array mixer	10
	2x3-array mixer	6
	2x4-array mixer	3
	4-electrode linear array mixer	5
Opt	LED+Photodiode	30
Storage	Single cell	N/A

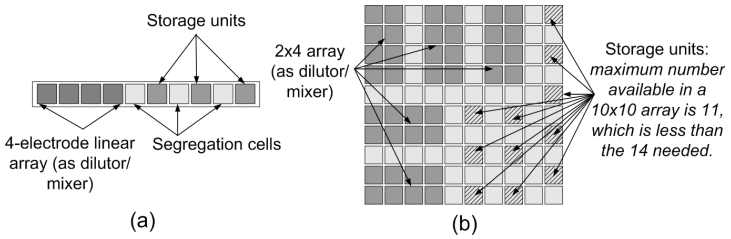

Figure 5. (a) Baseline case 1: a full-custom design; (b) Baseline case 2: violation of given specifications for the design obtained from [4].

operations of dilution, mixing and optical detection have to be carried out sequentially. Consequently, the completion time for the protein assay is as high as 560 seconds, which exceeds the design specification of 400 seconds. As a second baseline case, we attempt to minimize the assay processing time using the genetic algorithm that was proposed in [4]. In this method, only area estimates of the microfluidic array, i.e., the sum of the areas of active microfluidic modules in each time step, are used to guide the scheduling procedure. To minimize the operation time, 2×4-array modules are used for dilution and mixing. A completion time of 297 seconds for the protein assay is obtained using this method. However, due to the absence of exact placement information, this design cannot guarantee that spatial constraints on the design are satisfied. For example, it can be shown that at time step 167 seconds in this schedule, five 2×4-array dilutors as well as 14 storage units are active on the array simultaneously. Although their area estimate satisfies the resource constraint (i.e., 5×8+14 = 54 < 10×10), we cannot pack these microfluidic modules without overlaps in a 10×10 array if we incorporate the segregation regions between modules, as shown in Figure 5(b). This implies that the resulting design fails to meet the design specification related to array area.

We now use the PRSA-based algorithm described in Section 4.2 to find a desirable solution that satisfies design specifications. In the simulation experiments, we set the number of chromosomes in the population to 103. During evolution, the 10 best chromosomes are reproduced into the next generation. A total of 36 chromosomes in the new population result from the crossover. The remaining 57 chromosomes are obtained from the mutation operators, where 19 new chromosomes are from the mutation of genes involved with resource binding, 19 from the mutation of genes for scheduling, and 19 from the mutation of genes for placement. Here mutation is implemented by randomly generating the new random keys to replace the old ones. For the annealing scheme, the initial temperature is chosen to ensure that almost every new child chromosome can be accepted in the Boltzmann trial, i.e., T_∞ = 10000. In the annealing process, the temperature is modulated as $T_{new} = k \times T_{old}$, where k = 0.9. The number of iterations of the inner loop for a given value of T is set to 5. In the objective function of the optimization problem, we set α = 0.5.

The unified synthesis method takes 110 minutes of CPU time on a 1.0 GHz Pentium-III PC with 256 MB of RAM. The solution thus obtained yields a biochip design with a 9×9 microfluidic array and the completion time for protein assay is 363 seconds. We illustrate the synthesis results, i.e., assay operation schedule and module placement, using a 3-D box model shown in Figure 6(a). Each microfluidic module is represented as a 3-D box, the base of which denotes the rectangular area of the module and the height denotes of the time-span of the corresponding assay operation. 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 all these boxes are contained in a bin of size

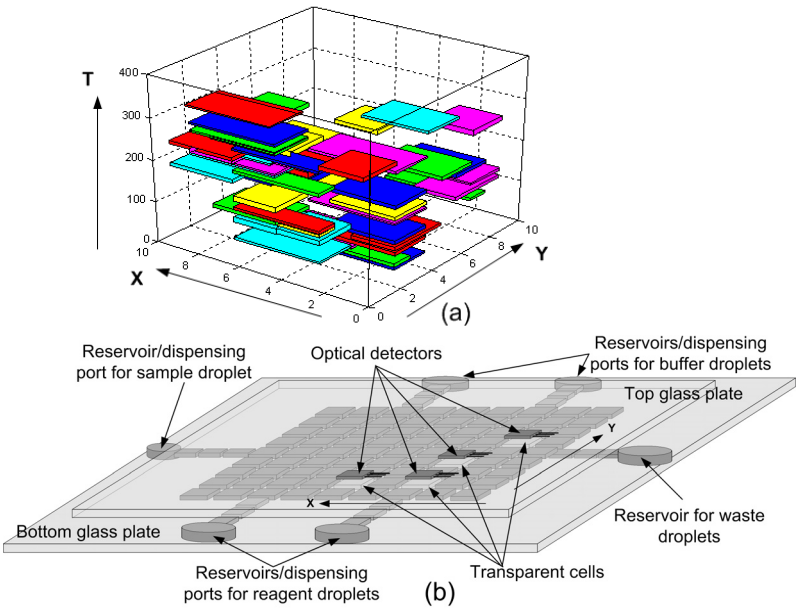

Figure 6. (a) A 3-D model illustrating the synthesis results; (b) a digital microfluidic biochip for a protein assay.

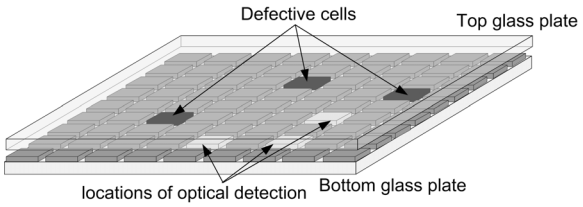

Figure 7. A defective 9×9 microfluidic array.

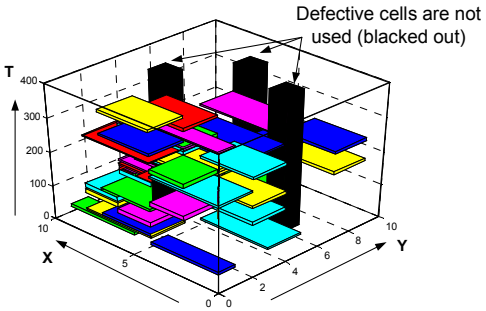

Figure 8. A 3-D model illustrating the reconfiguration results.

$X_{max} \times Y_{max} \times T_{max}$, where $A_{max} = X_{max} \times Y_{max}$; this implies that this design satisfies the specifications of array area and assay completion time. Moreover, there is no overlap between these boxes, thereby avoiding a violation of resource constraints. In addition, the synthesis results also determine the locations of integrated optical detectors. Transparent electrodes for optical detection are used in the microfluidic array. As shown in Figure 6(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 the protein assay.

Next we investigate defect tolerance using the above example. Assume that the above biochip has been fabricated. Suppose that due to particle contamination, some cells in this 9×9 microfluidic array are rendered defective; an example is shown in Figure 7. In order to ensure that the protein assay can still be carried out on this biochip, we need to bypass these faulty cells during assay operation. Moreover, due to defective cells, some non-reconfigurable resources may no longer be available. In this example, we assume one that optical detector is rendered defective after manufacturing. Thus the operations assigned to this detector have to be remapped to other detectors. The modified synthesis method proposed in Section 4.3 is used here to carry out the reconfiguration to tolerate these manufacturing defects. The reconfiguration results are shown in Figure 8. This new design, allows the protein assay to operate on this defective biochip with an increase of only 6% in the completion time, i.e., the completion time is now 385 seconds.

6. CONCLUSION

We have presented a new synthesis methodology for droplet-based microfluidic biochips. The synthesis procedure, which is based on parallel recombinative simulated annealing, unifies the scheduling of bioassay operations, resource binding, and module placement. We have also shown that the proposed synthesis method can be used after fabrication to tolerate manufacturing defects. The real-life example of a protein assay based on the Bradford reaction has been used to evaluate the effectiveness of the synthesis procedure. This work is expected to facilitate the automated design of biochips; in this way, the biochip user can concentrate on the development of nano- and micro-scale bioassays, leaving implementation details to the synthesis tools. This will in turn pave the way for the integration of biochip components in the next generation of system-on-chip designs, as envisaged by the 2003 ITRS document.

7. REFERENCES

[1] R. B. Fair et al., "Electrowetting-based on-chip sample processing for integrated microfluidics", *Proc. IEDM*, pp. 32.5.1-32.5.4, 2003.

[2] International Technology Roadmap for Semiconductors (ITRS), http://public.itrs.net/Files/2003ITRS/Home2003.htm.

[3] J. Zeng and T. Korsmeyer, "Principles of droplet electrohydrodynamics for lab-on-a-chip", *Lab on a Chip*, pp. 265-277, 2004.

[4] F. Su and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips", *Proc. ICCAD*, pp. 223-228, 2004.

[5] G. De Micheli, *Synthesis and Optimization of Digital Circuits*. New York: McGraw-Hill, 1994.

[6] W. E. Dougherty and D. E. Thomas, "Unifying behavioral synthesis and physical design", *Proc. DAC*, pp. 756-760, 2000.

[7] K. Bazargan et al, "Integrating scheduling and physical design into a coherent compilation cycle for reconfigurable computing architectures", *Proc. DAC*, pp. 635-640, 2001.

[8] I. Koren and A. Singh, "Fault tolerance in VLSI circuits", *IEEE Computer*, vol. 23, pp. 73-83, 1990.

[9] G. Li and N. Aluru, "Efficient mixed-domain analysis of electrostatic MEMS", *IEEE Trans. CAD*, vol. 22, pp. 1228-1242, 2003.

[10] T. Mukherjee and G. K. Fedder, "Design methodology for mixed-domain systems-on-a-chip [MEMS design]", *Proc. IEEE VLSI System Level Design*, pp. 96-101, 1998.

[11] S. K. Tewksbury, "Challenges facing practical DFT for MEMS", *Proc. Defect and Tolerance in VLSI Systems*, pp. 11-17, 2001.

[12] T. Zhang et al., *Microelectrofluidic Systems: Modeling and Simulation*, CRC Press, Boca Raton, FL, 2002.

[13] V. Srinivasan et al., "Protein stamping for MALDI mass spectrometry using an electrowetting-based microfluidic platform", *Proc. SPIE*, vol. 5591, pp. 26-32, 2004.

[14] M. Garey and D. Johnson, *Computers and Intractability-a Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

[15] S. W. Mahfoud and D. E. Goldberg, "Parallel recombinative simulated annealing: a genetic algorithm," *Parallel Computing*, vol. 21, pp. 1-28, 1995.

[16] J. C. Bean, "Genetics and random keys for sequencing and optimization", *ORSA J. Computing*, vol. 6, pp. 154-160, 1994.