Simultaneous Design and Placement of Multiplexed Chemical Processing Systems on Microchips

Anton J. Pfeiffer
Dept. Chemical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
pfeiffer@cmu.edu

Tamal Mukherjee
Dept. Electrical and Computer Eng.
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
tamal@ece.cmu.edu

Steinar Hauan
Dept. Chemical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
hauan@cmu.edu

Abstract

Microchip structures represent an attractive platform for microscale chemical processing of fluidic systems. However, standardized design methods for these devices have not yet been developed. Here we describe our work toward adapting traditional SoC circuit design techniques for the synthesis of fully customized and multiplexed Lab-on-a-Chip (LoC) devices. We discuss our formulation of the multiplex layout problem and present an approach for the design of microchip based electrophoretic separation systems. This work is extendable to systems incorporating mixing and reaction.

Keywords

Lab-on-a-Chip, fluidic circuit design, simultaneous design and placement

INTRODUCTION

Microchips represent a highly effective platform for the fabrication of microscale chemical sensors and analytical devices. Here we explore the possibility of adapting system on a chip (SoC) techniques for integrated circuit design to the design of complex microchip-based chemical analysis devices. Specifically we will be investigating devices known as $\mathrm{Lab\ on\ a\ Chip}$, (LoC) [1].

A LoC is essentially a miniaturized, integrated version of a macroscale analytical chemistry laboratory constructed in a microchip structure. LoC's have the potential to be efficient, automatable, portable, disposable and inexpensive to fabricate [2]. They have already seen a great deal of use within the life-science and biomedical industries for applications in genomics, proteomics and combinatorial chemistry [1]. LoC devices have contributed toward the recent advances in genome mapping. Future uses include non-invasive blood glucose measurement and even implantation into living tissue for monitoring and therapeutic purposes [3].

Recently, partitioning based approaches have been successfully used to design regular-arrays of DNA probes on microchips [4]. DNA arrays can be thought of as a subset or possible subsystem within an LoC device. For instance, in the case of DNA analysis, our methodology would allow for the simultaneous design and layout of the channel network that would be needed to sample body fluid, biochemically pre-process the fluid, and bring the analyte to the DNA sensor array. Our architecture is extendable to full-custom LoC design.

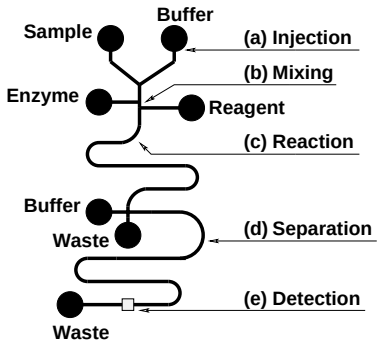

Figure 1. Canonical Lab-on-a-Chip

BACKGROUND

The fabrication of LoC devices is typically done on glass or plastic substrates with techniques adapted from the semiconductor industry. Figure 1 represents our canonical view of the major components of a LoC device. Sample injection (a), mixing (b), reaction (c), separation (d) and detection (e) all take place within a single microchip. Of the chemical processes implemented on microchips, separation, specifically capillary electrophoresis (CE) has been shown to separate many important biological molecules and inorganic compounds [5]. In this paper, we will demonstrate our method by automatically designing chips containing many CE subsystems. Although we focus on CE devices, our method is extendable to the incorporation of on-chip reaction, mixing and injection.

Figure 2 shows the major components of a simple chip based capillary electrophoretic separation system. The chip is composed of: (a) an injector where the analyte $(\mathrm{mixture\ of\ chemical\ species\ to\ be\ separated})$ enters the system, (b) the separation channel where the analyte separates into unique species bands and, (c) the detector where species bands are detected, (typically optically or electrically). Electrodes positioned in the four wells generate an electric field that drives both the separation speed and species travel direction.

Electrophoretic separation occurs because of the differential transport of charged species in the presence of an electric field. As an analyte mixture travels through an electrophoretic channel, the species within the mixture separate into bands according to their electrophoretic mobilities. While separation occurs, the species bands broaden or disperse due to factors such as diffusion, geometry, Joule heat-

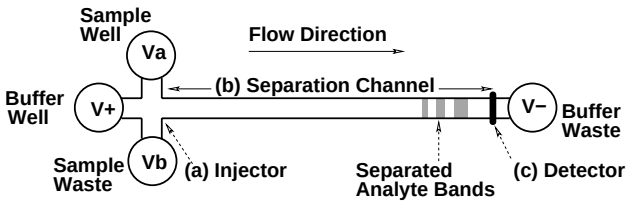

Figure 2. Simple chip-based CE schematic.

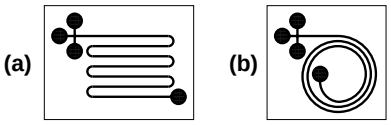

Figure 3. Serpentine (a) and spiral (b) topologies.

ing, adsorption, and electromigration [6]. A performance metric that represents the ratio of the distance between the centers of two adjacent bands to the average dispersion of the bands is termed resolution $(\hat{R})$.

In traditional CE, when an increase in separation performance is required, the separation channel is lengthened. However, long channels can only be made to fit on a microchip by adding turns to the design, which results in a phenomena called turn induced dispersion. The topology of a design, or interconnectivity of channel sections, can also be shown to have a significant impact on the performance of a design [7]. Thus, the minimization of design area and the maximization of separation performance represent conflicting design goals. Two common topologies found in the literature are the serpentine [8] and spiral [9] topologies (Fig. 3a and Fig. 3b). We focus on the serpentine topology because this topology facilitates easy access to I/O ports in a multiplexed layout.

Multiplexed Chips

The most complex multiplexed fluidic microchips to date consist of arrays, where hundreds of replicated simple channel structures function in parallel [10]. The typical layout of these devices is in a spoked wheel configuration with straight separation channel sections making up the spokes. These devices benefit from a high degree of parallelism. This allows for fast analysis of combinatorial experiments, as well as a high degree of redundancy which is important for precise experimental reproducibility. However, this simple topology results in designs that are very large when applied toward difficult separations, such as the separation of DNA sequences that may differ by only a few base pairs. Furthermore, a topology composed of identical straight sections is not capable of handling experimental protocols in which a complex series of sequential and parallel assays is required.

In our approach, we tackle these problems by multiplexing serpentine topologies. Each subsystem design is defined by a set of chemical properties and design specifications. Unlike the spoked-wheel configuration, our multiplexed designs are composed of a heterogeneous collection of serpentine subsystems, each designed to perform a specific separation. A single multiplexed device can simultaneously perform many different difficult separations in a compact chip area.

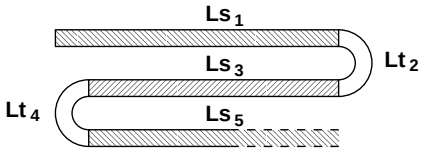

Figure 4. Decomposition of serpentine topology

Design Issues

In LoC devices, the subsystem design and ultimate chip layout are intimately linked due to the influence that channel geometry has on device performance. Therefore, a solution approach that is capable of simultaneously considering both subsystem design and overall system layout is required.

We are not aware of any standard methodologies for the design of multiplexed LoC devices. Currently, multiplexed designs require a substantial investment of time, manpower and expertise. Typical design cycles often include extensive laboratory experimentation, time consuming numerical simulations and manual verification processes. The adaptation of SoC techniques for chip layout coupled with an effective methodology for subsystem design, has the potential to reduce the length of design cycles to only days and to facilitate the creation of innovative applications for LoC technology.

Optimal Serpentine Design

Channel topologies can be decomposed into sections such as straights, elbows, and U-bends. Figure 4 shows a decomposition of a serpentine topology. We define a particular instance of a serpentine topology to be a vector $\mathcal{T}$, where the odd elements are straight section lengths (Ls), and the even elements are turn lengths (Lt) along the center-line radius of the channel $(\mathcal{T} = [Ls_1, Lt_2, Ls_3 \ldots])$.

An electrophoretic channel system can be simulated by piecing the channel sections together to produce the desired channel topology. We have created an electrophoretic channel simulator that employs accurate algebraic physical models combined with logic [11]. We are currently able to construct and simulate the vast majority of the topologies in the literature. Equation 1 is a simplified representation our electrophoretic channel simulator which shows the information relevant to the current problem.

$$[X, Y, \hat{R}, E] = SIMULATOR(\mathcal{T}, V, props \ldots) \quad (1)$$

The simulator takes in the topology instance $\mathcal{T}$, the operating voltage, V, and an object containing the system chemical properties. The simulator returns the dimensions of the design, X and Y, the separation performance, $\hat{R}$, and the electric field strength, E.

In previous work [7], we developed a nonlinear program (NLP) formulation of the serpentine separation channel prob-

lem summarized below (2).

$$\begin{aligned} \min. \quad & Area_{(\mathcal{T})} = X \cdot Y \quad \forall \mathcal{T} \in \boldsymbol{S} \\ \text{st.} \quad & \hat{R}_{m,n} \geq \hat{R}_{spec} \quad \forall m, n \in props \\ & E \leq E_{max} \\ & V \leq V_{max} \\ & w \leq Ls \leq X \\ & \pi \cdot r_{min} \leq Lt \leq \frac{\pi \cdot Y}{2} \\ & \pi \cdot PAD \leq Lt \end{aligned} \tag{2}$$

In this formulation, the objective is to minimize the area of a given topology $Area_{(\mathcal{T})}$. The bounding box of each subsystem is defined by X and Y. The resolution, $\hat{R}_{m,n}$, between species m and n in each subsystem must be greater than a user specified performance value, $\hat{R}_{spec}$. Furthermore, the electric field strength, E, and the applied voltage, V, must be less than the operational specifications, E_{max} and V_{max}, respectively. Bounds on the lengths of straight sections, Ls, and the lengths of turns, Lt, are also invoked. The acceptable region of model validity sets the value of the minimum turn radius, r_{min}, and the fabrication method defines the inter-channel spacing, PAD. All of the constraints in (2) are incorporated into our multiplexed system design methodology. The objective function in (2) is replaced by a total system objective function that will be discussed in the formulation section.

Combining (2) with a layout problem adds a level of complexity to an already difficult problem and results in a multilevel optimal design problem. We are currently researching methods to reduce the complexity of our original NLP [12].

PLACEMENT PROBLEM DEFINITION

Equations (1) and (2) form the basis for the design of each individual subsystem. An example subsystem is shown in Fig. 5. The coordinate location of the sample port (S^x, S^y), buffer port (B^x, B^y), sample waste (Sw^x, Sw^y), buffer waste (Bw^x, Bw^y) and subsystem dimensions, X and Y, are labeled. We assume that all subsystems will be similar to the topology shown in Fig. 5 with ports deterministically located one per side. This configuration allows for convenient subsystem I/O and heuristically reduces the potential for routing congestion around each subsystem.

Figure 6 is an example instance of a design with three subsystems. Auxiliary channels (a) transport fluid between subsystem ports and particular wells (b). The wells form the world-to-chip interface and allow fluids to be introduced and removed from the chip. The position of each subsystem and well is defined by an (x, y) coordinate point in its lower left corner. Notice that we model wells as squares with inscribed circles. Figure 5 and Fig. 6 present a graphical description of the simultaneous subsystem design and layout problem.

We define the problem more formally as follows: given a set of subsystems, $\mathcal{N}$, and their associated input and output

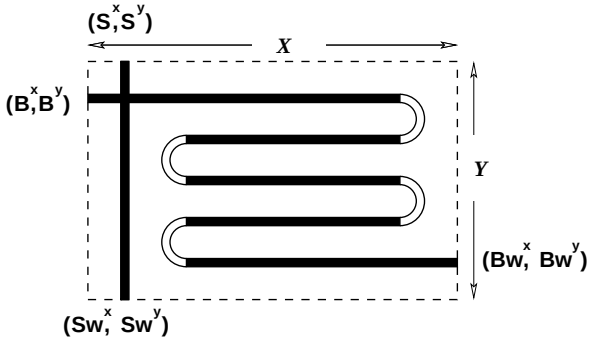

Figure 5. Subsystem schematic

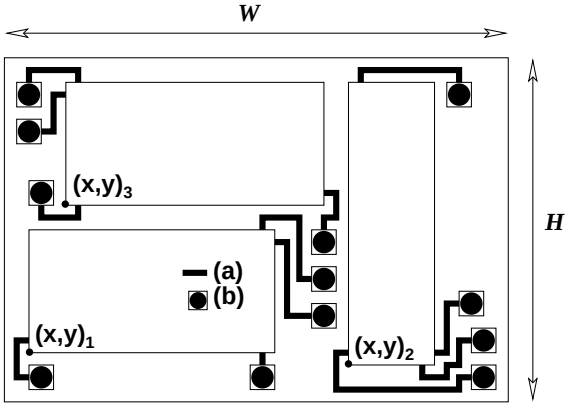

Figure 6. Chip layout instance

wells, $\mathcal{W}$, we attempt to create a planar design where no dimension exceeds the total chip height H, or width W and the occupied chip area is compact. All of the subsystems must meet the required performance specifications. Additionally, we want to place wells as close as possible to the input and output ports of their associated subsystems, minimizing the auxiliary channel length required to move analytes on the chip.

It is apparent that our problem is very similar to a standard VLSI placement problem in which chip area and wire length are to be minimized. We assume a building-block layout style for our problem as opposed to a grid-point assignment layout style [13] because little information about subsystem size is known a priori and subsystems may have highly variable dimensions. However, unlike many typical VLSI placement formulations, subsystem aspect ratio, or total area is not explicitly known. Subsystem dimensions are a function of subsystem design specifications such as bounds on separation performance $\hat{R}$. The subsystem dimensions and total system layout are determined simultaneously to produce a minimal area design that meets all subsystem performance specifications.

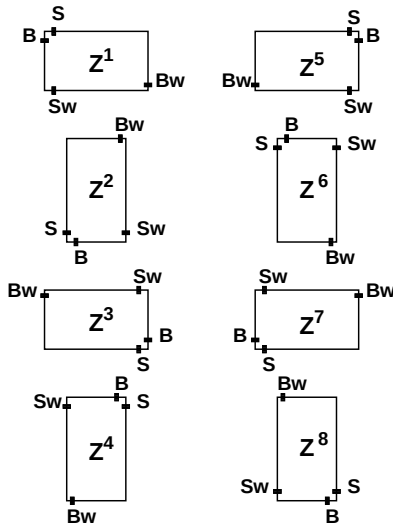

Figure 7. Eight possible subsystem orientations

PROBLEM FORMULATION

We develop a rigorous problem formulation using a concise modeling technique known as General Disjunctive Programming (GDP) [14]. GDP can be effectively translated into mixed integer nonlinear programs (MINLP) and solved using standard algorithms.

We define i as a unique subsystem in $\mathcal{N}$, and j as a unique well in $\mathcal{W}$. We also define an ordered composite set $\mathcal{K}$ as the union of wells and subsystems, $\mathcal{K} = \mathcal{N} \cup \mathcal{W}$, with k being a unique subsystem or well in $\mathcal{K}$. This composite set represents the complete set of blocks to be placed on the chip. We divide the formulation into 4 parts: P1 - Subsystem rotation, P2 - Subsystem overlap, P3 - Well to port assignment, and P4 - General constraints and objective.

Subsystem Rotation (P1)

In (3) we define the 8 possible orientations and associated port locations for each subsystem i in $\mathcal{N}$. The 8 orientations are rotations and reflections of the serpentine subsystem shown in Fig. 5.

$$\left.\begin{bmatrix} Z_i^r \\ S_i = (S^x, S^y) \\ Sw_i = (Sw^x, Sw^y) \\ B_i = (B^x, B^y) \\ Bw_i = (Bw^x, Bw^y) \end{bmatrix} \begin{array}{c} \vee \\ \forall r = 1 \ldots 8 \end{array} \right\} \forall i \in \mathcal{N} \qquad (3)$$

In (3), one of the boolean variables, Z_i^1 through Z_i^8 will be assigned $True$ depending on it's orientation as shown in Fig. 7. For convenience, each boolean variable can be assigned a binary value, $True = 1$, $False = 0$. The coordinates of the sample (S_i), sample waste (Sw_i), buffer (B_i), and buffer waste (Bw_i) ports are determined for an orientation of a subsystem with respect to the bottom left-hand corner of a particular block.

We model the set of wells, $\mathcal{W}$, as squares, $(X_j = Y_j)$, superscribing circular wells (Fig. 6). The coordinate point, W_j in (4), represents a point in the center of well j where auxiliary channels can connect. Hence, the rotational orientation of wells does not have to be considered and we can define a ninth binary, Z_j^9, and set its value to 1 for all wells.

$$\begin{array}{c} Z_j^9 = 1 \\ W_j = (W^x, W^y) \end{array} \quad \forall j \in \mathcal{W} \qquad (4)$$

Equation (4) allows us to introduce the set of wells into $\mathcal{K}$ and to generalize the formulation of block overlap prevention.

Overlap Prevention (P2)

Since we assume a building block layout style, blocks can be placed anywhere on the chip, but a legal placement is only achieved when no blocks overlap. Overlap is prevented by assigning $\mathrm{right}, \mathrm{left}, \mathrm{above}, \mathrm{below}$ relationships between each block in $\mathcal{K}$, $(l, k \in \mathcal{K})$. Here we show a GDP that enforces these relationships for k left of l (5), k right of l (6), k below l (7) and k above l (8).

$$\begin{bmatrix} \delta_{k,l}^1 \\ x_k + X_k \sum_{r=0}^{9/2} Z_k^{2r+1} + Y_k \sum_{r=1}^{9/2} Z_k^{2r} \leq x_l \end{bmatrix} \vee \qquad (5)$$

$$\begin{bmatrix} \delta_{k,l}^2 \\ x_k - X_k \sum_{r=0}^{9/2} Z_k^{2r+1} - Y_k \sum_{r=1}^{9/2} Z_k^{2r} \geq x_l \end{bmatrix} \vee \qquad (6)$$

$$\begin{bmatrix} \delta_{k,l}^3 \\ y_k + X_k \sum_{r=0}^{9/2} Z_k^{2r+1} + Y_k \sum_{r=1}^{9/2} Z_k^{2r} \leq y_l \end{bmatrix} \vee \qquad (7)$$

$$\begin{bmatrix} \delta_{k,l}^4 \\ y_k - X_k \sum_{r=0}^{9/2} Z_k^{2r+1} - Y_k \sum_{r=1}^{9/2} Z_k^{2r} \geq y_l \end{bmatrix} \qquad (8)$$

$$ord(k) < ord(l), \quad \forall k, l \in \mathcal{K}$$

The values of the binary variables $\delta_{k,l}^1$, $\delta_{k,l}^2$, $\delta_{k,l}^3$, and $\delta_{k,l}^4$ determine the orthogonal packing of the blocks. A disjunction is active when its associated binary equals one. The summation terms in (5 - 8) account for the possibility of block rotation by using the binary Z_k^r variables from P1 to correctly determine the orientation of block dimensions X_k and Y_k. It should be noted that since subsystem dimensions, X_k and Y_k are not predefined, the equations in P1 and P2 are nonlinear upon relaxation and therefore more difficult to solve.

Well to Port Assignment (P3)

Here we model the connection between subsystem ports and wells as an assignment problem. We assume that there is a unique one to one mapping between ports and wells. Our

goal is to minimize the distance that auxiliary channels must span to connect a port to a well. Constraints (9) enforce that a port of a particular subsystem is assigned to only one well.

$$\begin{aligned} \sum_{j\in\mathcal{W}} PS_{i,j} = 1 \qquad & \sum_{j\in\mathcal{W}} PSw_{i,j} = 1 \\ \sum_{j\in\mathcal{W}} PB_{i,j} = 1 \qquad & \sum_{j\in\mathcal{W}} PBw_{i,j} = 1 \end{aligned} \quad \forall i \in \mathcal{N} \qquad (9)$$

The binary variables, $PS_{i,j}, PSw_{i,j}, PB_{i,j}$ and $PBw_{i,j}$ represent the connection of the sample, sample waste, buffer and buffer waste ports to a particular well. Constraint (10) enforces that a well is connected to a single port.

$$\sum_{i\in\mathcal{N}} (PS_{i,j} + PSw_{i,j} + PB_{i,j} + PBw_{i,j}) = 1 \qquad (10)$$

$$\forall j \in \mathcal{W}$$

We use a rectilinear distance metric [13] to estimate the length that an auxiliary channel must span between a port and a well. This connection length is captured by the variables $CS_{i,j}, CSw_{i,j}, CB_{i,j}$ and $CBw_{i,j}$ for a given port to well connection as shown in (11).

$$\begin{aligned} CS_{i,j} &= |S_i^x - W_j^x| + |S_i^y - W_j^y| \\ CSw_{i,j} &= |Sw_i^x - W_j^x| + |Sw_i^y - W_j^y| \\ CB_{i,j} &= |B_i^x - W_j^x| + |B_i^y - W_j^y| \\ CBw_{i,j} &= |Bw_i^x - W_j^x| + |Bw_i^y - W_j^y| \end{aligned} \qquad (11)$$

The equations in (11) are trivially reformulated into a system of linear inequalities which avoids discontinuities in the Jacobian, and are only written using absolute value to be concise.

General Constraints and Objective (P4)

The design region is restricted to the chip space available, defined by H and W. Therefore, constraints are necessary to prevent any of the subsystems or wells from extending outside of the design region. The following constraints (12) enforce that all blocks are placed within the design region.

$$\begin{aligned} x_k + X_k \sum_{r=0}^{9/2} Z_k^{2r+1} + Y_k \sum_{r=1}^{9/2} Z_k^{2r} \le W \\ y_k + X_k \sum_{r=0}^{9/2} Z_k^{2r+1} + Y_k \sum_{r=1}^{9/2} Z_k^{2r} \le H \end{aligned} \qquad (12)$$

The bounding box of the design can be determined similarly. Equations (13) determine the right, R, left, L, top, T, and bottom, B, edges of the design.

$$\begin{aligned} x_k + X_k \sum_{r=0}^{9/2} Z_k^{2r+1} + Y_k \sum_{r=1}^{9/2} Z_k^{2r} \le \mathrm{R} \\ x_k - X_k \sum_{r=0}^{9/2} Z_k^{2r+1} - Y_k \sum_{r=1}^{9/2} Z_k^{2r} \ge \mathrm{L} \\ y_k + X_k \sum_{r=0}^{9/2} Z_k^{2r+1} + Y_k \sum_{r=1}^{9/2} Z_k^{2r} \le \mathrm{T} \\ y_k - X_k \sum_{r=0}^{9/2} Z_k^{2r+1} - Y_k \sum_{r=1}^{9/2} Z_k^{2r} \ge \mathrm{B} \end{aligned} \qquad (13)$$

Our objective function is the weighted sum of the bounding box perimeter and the total auxiliary channel length (14). The weights, α_1 and α_2, can be assigned values to emphasize either the area minimization or the auxiliary channel length minimization.

$$\begin{aligned} \min. \quad F = \alpha_1[(\mathrm{R} - \mathrm{L}) + (\mathrm{T} - \mathrm{B})] \\ + \alpha_2 \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{W}} \{CS_{i,j}PS_{i,j} + CSw_{i,j}PSw_{i,j} \\ + CB_{i,j}PB_{i,j} + CBw_{i,j}PBw_{i,j}\} \end{aligned} \qquad (14)$$

Notice again that the relaxed form of (12) and (13) are nonlinear. Also the cost matrices, $CS_{i,j}$, $CSw_{i,j}$, $CB_{i,j}$, and $CBw_{i,j}$ are not static and therefore the relaxed form of the objective is also nonlinear.

The equations in P1, P2, P3, P4 and (2) complete our description of the problem. In the next sections we discuss a method to simultaneously determine the design of each subsystem and it's placement on the chip.

Problem Size Reduction

While we consider our formulation to be a rigorous description of the microchip CE placement problem, it is impractical to solve this formulation directly. MINLPs of this size and complexity can only be solved for trivial cases (i.e. $|\mathcal{N}| \leq$4). It is possible to reformulate the equations in P1, P2, P3, and P4 into a mixed integer linear program (MILP). However, this requires the addition of a substantial number of constraints and variables (i.e. $> 10^4$ equations when $|\mathcal{N}|$=6). Furthermore, the underlying physics contained in the subsystem models is highly nonlinear and can not be readily linearized. Since no general methods are capable of directly handling a problem of this complexity, we examine ways to reduce the problem complexity and then create a tailored solution approach.

One possible way to simplify the formulation is to place the wells along the chip edges (Fig. 8). In the most general case, we require four wells for each subsystem. The ability to pre-place the wells in a defined region allows us to remove $\mathcal{W}$ from $\mathcal{K}$ in P2 and reduce the problem size by $4|\mathcal{N}|$. Placing wells on the chip boundary is a reasonable physical assumption because it eliminates the $\mathrm{world\ to\ chip}$ interfac-

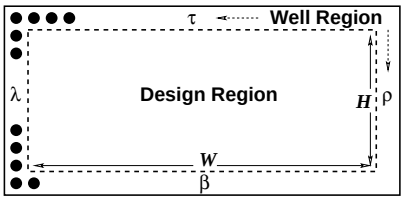

Figure 8. Simplification by creating well and design regions

ing issues that would arise if wells were placed throughout the chip.

The placement of wells on chip edges can be described by a disjunction (15).

$$\begin{bmatrix} \lambda_j \\ W_j^x + X_j = 0 \end{bmatrix} \vee \begin{bmatrix} \rho_j \\ W_j^x = W \end{bmatrix} \vee \qquad (15)$$

$$\begin{bmatrix} \tau_j \\ W_j^y = H \end{bmatrix} \vee \begin{bmatrix} \beta_j \\ W_j^y + Y_j = 0 \end{bmatrix}$$

The boolean variables, λ_j, ρ_j, τ_j, and β_j assign well j to either the left, right, top or bottom edge of the chip respectively (Fig. 8). This disjunction coupled with an appropriate overlap prevention constraint for wells along an edge allows for the legal placement of wells on the chip boundary.

Problem P3 can be eliminated by defining a $\mathrm{Netlist}$ object [13], common in circuit design, that contains the port to well connectivity for each system.

SOLUTION METHOD

Despite the problem simplifications discussed above, the placement problem is well known to be NP-hard. Direct application of Branch and bound [15] and conventional MILP approaches [16] have been shown to be incapable of handling placement problems of a realistic size. Since MINLP solution strategies employ both of these methods, our formulation can not be solved directly and a tailored solution method is necessary.

Several interesting approaches have been presented in the VLSI literature to solve the placement problem. Many of these approaches rely on creating a problem representation that can be efficiently searched by a probabilistic heuristic such as simulated annealing (SA) or genetic algorithms (GA). We have chosen to adapt a non-slicing representation known as Sequence Pair (SP) [17] to our problem. We have chosen the SP representation because of its generality and flexibility. When combined with an efficient Longest-common-subsequence (LCS) algorithm [18], SP has been shown to be competitive with other recent general placement methods.

Figure 9 is a flowchart illustrating our solution approach for the simultaneous design and layout problem. The main idea of this approach is to use a probabilistic search heuristic such as SA, to deal with the combinatoric aspects of the problem and an efficient gradient based approach for the remaining continuous-space problem. The integer variables in our formulation are instantiated by the heuristic resulting in a NLP. New NLPs are dynamically constructed for each new problem instance. This approach is conceptually similar to placement methods meant to handle soft blocks, where the resulting instantiated problem is an LP [19] or a convex program [20]. In our case, we are not only concerned with placement, but also with subsystem design, full block rotation, netlist generation and constraint satisfaction.

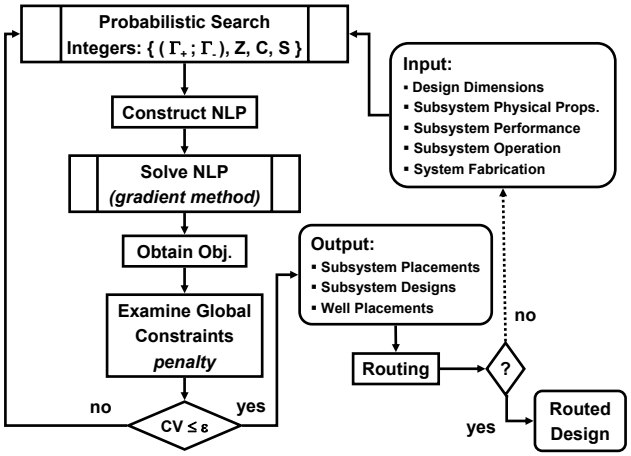

Figure 9. Probabilistic instantiation algorithm

The algorithm in Fig. 9 begins by obtaining the relevant subsystem and chip design specifications. Next, the search heuristic proposes a sequence pair, $(\Gamma_+;\Gamma_-)$, an instance of the block rotation vector, $Z = [z_1, z_{|\mathcal{M}|}]$ where $z_i \in \{1 \ldots 8\}$, an instance of the port-well connectivity vector $C = [c_1, c_{|\mathcal{W}|}]$ where each c_j is a unique number between 1 and $|\mathcal{W}|$, and an instance of the subsystem topology vector, $\boldsymbol{S} = [\mathcal{T}_1, \mathcal{T}_{|\mathcal{M}|}]$. From this information, an NLP can be dynamically constructed by: (a) mapping Z to the appropriate binaries in P1, (b) using $(\Gamma_+;\Gamma_-)$ to determine the relative placement of blocks in P2, (c) mapping C to the appropriate binaries in P3 and (d) solving each subsystem design in $\boldsymbol{S}$, simultaneously. The solution of the NLP results in the optimal design of the particular instance. We solve the NLP for our original objective, (14), but we do not include the chip boundary constraints in (12) to prevent the inner-loop NLP from becoming infeasible. Instead, we consider these constraints $\mathrm{global\ constraints}$ and handle them using a penalty function approach outside of the NLP. We consider the problem to be converged when the constraint violation, CV, on the global constraints is below a tolerance, ϵ. If the constraints are satisfied, the placement and subsystem information is output. Currently, our method does not handle routing. We are working on incorporating a planar single layer routing procedure into our method.

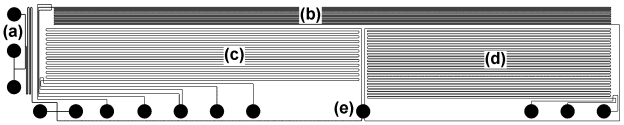

Figure 10. Multiplexed design schematic

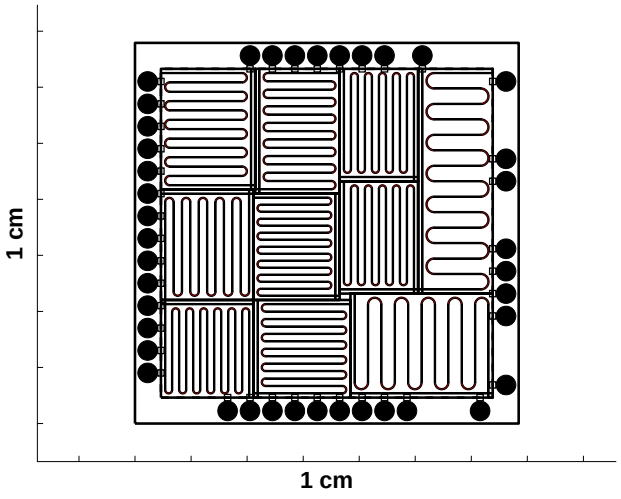

Figure 11. Compact placement of 10 subsystems

PLACEMENT RESULTS

Our early work on the design of LoC devices motivated our development of the formulation and solution method presented in this paper. The device shown in Fig. 10 is a multiplexed design composed of 4 subsystems designed to fit on a glass wafer measuring 9cm $\times$ 1.7cm. The subsystem designs were created based on heuristic methods for single system design [7]. The placement, routing and post-design verification were preformed by hand. Despite the simplicity of this design, the entire design process took over 5 hours.

Using the method described in this paper, we are able to obtain far more complex designs in far less time. Figure 11 shows a placed and compacted design including well positions for a multiplexed chip containing 10 subsystems. This design was produced in under 10 minutes of CPU time on a standard PC ($2GHz\ P4$, $1Gb\ ram$).

Our method is scalable and capable of handling designs with many more subsystems. Figure. 12 shows the average change in objective value ($\mathrm{dotted\ line}$) and the average function evaluation time for constructing and solving the NLP ($\mathrm{dash\ \&\ dotted\ line}$) versus the number of subsystems. The time-cost of constructing and solving the NLP is the dominant time-cost in our method. We therefore use it as the basis for our analysis.

We attempted to create a representative experiment by choosing typical physical property values, operating conditions and performance specifications for each subsystem. The analysis was performed for random configurations of designs con-

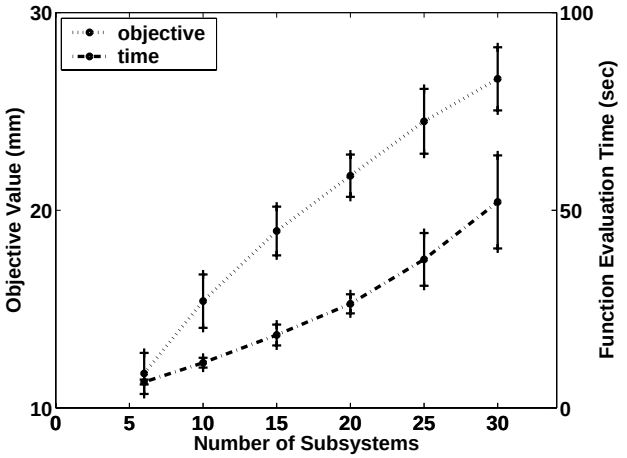

Figure 12. Function scaling per instance size

taining 6, 10, 15, 20, 25, and 30 subsystems. Each trial was performed 20 times on a standard PC ($1GHz\ P3$, $1Gb\ ram$). The mean value and one standard deviation above and below the mean are shown on the plot for both the function evaluation time and objective value.

As expected, the standard deviations of both time and objective value increase as the number subsystems increase. This is because the solution space grows in proportion to $(|\mathcal{N}|!)^2 \cdot 8^{|\mathcal{N}|}$. We use the mean and standard deviation data obtained from Fig. 12 to tune the cooling schedule of the SA algorithm used in our method (Fig. 9).

Although the worst-case time complexity of typical NLP solvers is exponential, our experiments indicate that the average time complexity of our algorithm is not strongly exponential. However, it is obvious that as problem instance size increases, solution time will become prohibitive, resulting in a loss of solution quality. To combat this problem, we are currently developing high-speed greedy initialization heuristics and parallelization techniques for our method.

CONCLUSION AND FUTURE WORK

We have presented a rigorous GDP formulation for the design of chip based multiplexed separation subsystems as well as a solution approach based on methods adapted from SoC circuit design. The SoC literature is a rich source of insight and we are currently investigating several alternative solution approaches for the placement problem. In addition, we are experimenting with a distributed agent system for the solution of this problem. The agent system was found to be very effective in earlier work [12].

We are formulating and developing solution approaches for the routing of ports to wells. Routing represents a major challenge in the synthesis of LoC devices. Auxiliary channels must be as short as possible to prevent sample nonuniformities from developing and to reduce the voltage required to drive the system. While the routing of LoC auxiliary channels is superficially similar to wire routing in VLSI

circuit design, there are several complicating factors. Most importantly, LoC devices are generally fabricated in a single layer to reduce fabrication costs and complexity. This means that all auxiliary channels must be routed in a planar fashion and can not be routed above or below a subsystem. Furthermore, the assumptions that channels can feed through a subsystem or that ports may move along a subsystem edge, do not apply. In addition, auxiliary channels occupy significant space on the chip and can not be assumed to be of insignificant width. This can lead to heavy congestion in the routing regions of a chip.

Some work has been presented on the single-layer routing problem [21]. We are currently examining the conventional circuit design approach of performing placement and routing in a hierarchical fashion. One simple approach is to surround each subsystem with a technology dependent routing halo [15, 17]. A post routing compaction phase can be employed to eliminate unused routing space. The other possibility is to employ a global routing procedure based on a multi-commodity flow model to determine the location of auxiliary channels [13]. Ideally, subsystem design, placement and routing will be handled simultaneously, as shown in Fig. 9.

Acknowledgment

This research effort is sponsored by the Defense Advanced Research Projects Agency (DARPA) and U. S. Air Force Research Laboratory, under agreement number F30602-01-2-0987 and by the National Science Foundation (NSF) under award CCR-0325344. The authors would like to thank members of the SYNBIOSYS group at Carnegie Mellon, especially Dan Paterson for his early ideas related to LoC layout, and Bikram Baidya for his suggestions.

REFERENCES

[1] D.R. Reyes, D. Iossifidis, A. Auroux, and A. Manz. Micro Total Analysis Systems. 1 introduction, theory, and technology. Anal.Chem., 74:2623–2634, June 2002.

[2] W. Ehrfeld, V. Hessel, and H. Lehr. Microreactors: New Technology for Modern Chemistry. Wiley-VCH, 2000.

[3] J. Robert. Continuous monitoring of blood glucose. Hormone Research, 57:81–84, 2002.

[4] A. B. Kahng, I. Măndoiu, S. Reda, X. Xu, and A.Z. Zelikovsky. Evaluation of placement techniques for dna probe array layout. In ICCAD '03, pages 262–269, November 2003.

[5] N.A. Lacher, K.E. Garrison, R.S. Martin, and S.M. Lunte. Microchip capillary electrophoresis / electrochemistry. Electrophoresis, 22:2526–2536, 2001.

[6] B. Gas and E. Kenndler. Dispersive phenomena in electromigration separation methods. Electrophoresis, 2000.

[7] A.J. Pfeiffer, T. Mukherjee, and S. Hauan. Design and optimization of compact microscale electrophoretic separation systems. Ind.Eng.Chem.Res., 43:3539–3553, 2004.

[8] S.C. Jacobson, R. Hergenroder, L.B. Koutny, R.J. Warmack, and J.M. Ramsey. Effects of injection schemes and column geometry on the performance of microchip electrophoresis devices. Anal.Chem., 66:1107–1113, April 1994.

[9] C.T. Culbertson, S.C. Jacobson, and M.J. Ramsey. Microchip devices for high-efficiency separations. Anal.Chem., 72:5814–5819, December 2000.

[10] C.A. Emrich, H. Tian, I.L. Medintz, and R.A. Mathies. Microfabricated 384-lane capillary array electrophoresis bioanalyzer for ultrahigh-throughput genetic analysis. Anal.Chem., 74:5076–5083, 2002.

[11] Y. Wang, Q. Lin, and T. Mukherjee. Analytical dispersion models for efficient simulation of complex microchip electrophoresis systems. In Proceeding of MicroTAS, pages 135–138, 2003.

[12] A.J. Pfeiffer, J.D. Siirola, and S. Hauan. Optimal design of microscale separation systems using distributed agents. In Proceedings of FOCAPD 2004, pages 381–384, 2004.

[13] T. Lengauer. Combinatorial algorithms for integrated circuit layout. Wiley-Teubner, 1990.

[14] R. Raman and I.E. Grossmann. Modeling and computational techniques for logic based integer programming. Computers and Chemical Engineering, 18(7):563–578, 1994.

[15] H. Onodera, Y. Taniguchi, and K. Tamaru. Branch-and-bound placement for building block layout. In 28th ACM/IEEE Design Automation Conf., pages 433–439, 1991.

[16] S. Sutanthavibul, E. Shragowitz, and J.B. Rosen. An analytical approach to floorplan design and optimization. IEEE Trans. on CAD, 10(6):761–769, June 1991.

[17] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI module placement based on rectangle-packing by the Sequence Pair. IEEE Trans. on CAD, 15(12):1518–1524, December 1996.

[18] X. Tang and D.F. Wong. Fast-SP: A fast algorithm for block placement based on Sequence Pair. ASP-DAC 2001, pages 521–526, 2001.

[19] J.G. Kim and Y.D. Kim. A linear programming-based algorithms for floorplanning in VLSI design. IEEE Trans. on CAD, 22(5):584–592, May 2003.

[20] H. Murata and E.S. Kuh. Sequence-pair based placement method for hard/soft/preplaced modules. In Int. Symp. Physical Design, pages 167–172, 1996.

[21] M. Sarrafzadeh, K. Liao, and C.K. Wong. Single-layer global routing. IEEE Trans. on CAD, 13(1):38–47, January 1994.