Synthesis of On-chip Control Circuits for mVLSI Biochips

Seetal Potluri*, Alexander Schneider†, Martin Hørslev-Petersen†, Paul Pop†, Jan Madsen†
* Xilinx Asia Pacific, Singapore
† Department of Applied Mathematics and Computer Science, Technical University of Denmark

***Abstract*—Microfluidic VLSI (mVLSI) biochips help perform biochemistry at miniaturized scales, thus enabling cost, performance and other benefits. Although biochips are expected to replace biochemical labs, including point-of-care devices, the off-chip pressure actuators and pumps are bulky, thereby limiting them to laboratory environments. To address this issue, researchers have proposed methods to reduce the number of off-chip pressure sources, through integration of on-chip pneumatic control logic circuits fabricated using three-layer monolithic membrane valve technology. Traditionally, mVLSI biochip physical design was performed assuming that all of the control logic is off-chip. However, the problem of mVLSI biochip physical design changes significantly, with introduction of on-chip control, since along with physical synthesis, we also need to (i) perform on/off-chip control partitioning, (ii) on-chip control circuit design and (iii) the integration of on-chip control in the placement and routing design tasks. In this paper we present a design methodology for logic synthesis and physical synthesis of mVLSI biochips that use on-chip control. We show how the proposed methodology can be successfully applied to generate biochip layouts with integrated on-chip pneumatic control.**

I. Introduction

Microfluidics-based biochips integrate different biochemical analysis functionalities on-chip, miniaturizing the macroscopic biochemical processes to a sub-millimeter scale. These microsystems offer several advantages over the conventional biochemical analyzers, e.g., reduced sample and reagent volumes, faster biochemical reactions, ultra-sensitive detection and higher system throughput, with several assays being integrated on the same chip [1]. Biochips are used in many application areas like in vitro diagnostics (point-of-care), drug discovery (high-throughput screening, hit characterization), biotechnology (process monitoring and development) and ecology (agriculture, environment, homeland security). During the last decade, a significant amount of work has been carried out on the individual microfluidic components as well as the microfluidic platforms. There are several types of biochip platforms, each having its own advantages and limitations [1].

In this paper, we focus on flow-based biochips, in which the microfluidic channel circuitry on the chip is equipped with integrated microvalves, that are used to manipulate more complex units such as mixers, micropumps, multiplexers etc., with several hundreds of such units being accommodated on a single chip. Analogous to its microelectronics counterpart, this approach is called microfluidic Very Large Scale Integration (mVLSI) [2]. Although the soft lithography technology used for fabricating flow-based biochips has advanced faster than Moore's law [3], allowing increasing miniaturization, the off-chip pressure actuators and pumps are bulky, thereby limiting them to laboratory environments. To address this issue, recent work [4]–[7] focuses on reducing the number of off-chip pressure sources, using on-chip pneumatic control logic circuits fabricated using three-layer monolithic membrane valve technology [5]. However, all of these existing papers lack a systematic methodology to automatically synthesize on-chip control circuits, deciding on which portion to go on-chip and which portion to remain off-chip, to facilitate physical design for microfluidic very large scale integration (mVLSI). It is important to perform on/off-chip control partitioning, on-chip control circuit design over the integration of on-chip control in the placement and routing of the entire biochip.

This paper presents a systematic methodology for synthesis of on-chip control circuits for mVLSI biochips. The next section presents the system model, where we also explain how mVLSI on-chip control works. Section III highlights the issues related to the on-chip control synthesis at different phases of the overall mVLSI biochip design. In this paper we assume that the partitioning between on-chip and off-chip control is performed manually. We propose techniques for the logic synthesis of on-chip control circuits and for taking into account the synthesized on-chip control logic during the placement and routing steps of the physical design. Section IV presents the experiments we have performed to evaluate our methodology, and the results thus obtained. Section V concludes the paper.

II. System model

Physically, the biochip can have multiple layers, but the layers are logically divided into two types: *flow layer* and the *control layer*. The basic building block of a biochip is a micromechanical valve, which restricts/permits the fluid flow, and hence is used to manipulate the fluid in the flow layer. The microvalve is controlled using an external air pressure source, through a control pin. Control pins provide access to the control layer. The flow layer is connected to a fluid reservoir, through a pump that generates the fluid flow.

Broadly speaking, there are two types of microvalves proposed in prior literature: normally open valves [8] and normally closed valves [5]. We consider the latter in this paper. Such a normally-closed valve is shown as a switch-like structure, in Fig. 1a, where the valve remains closed under normal condition when no pressure is applied (P_N), and when atmospheric pressure (P_A) is applied at the control

978-3-9815370-8-6/17/$31.00

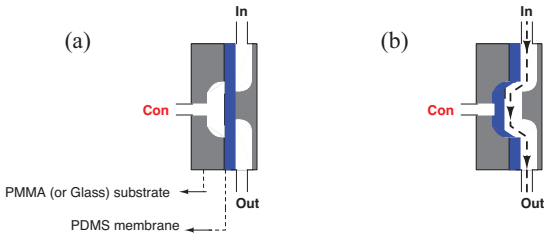


Figure 1. Basic switch used for on-chip control logic design, where In, Out and Con refer to Input, Output and Control ports respectively

signal (Con). The valve has three layers. The top and bottom layers are made of glass or polymethyl methacrylate (PMMA), and the middle layer is an elastic membrane made out of an elastomer such as polydimethylsiloxane (PDMS). The valve is normally closed, when vacuum is not applied, and "Con" has atmospheric pressure (P_A). When vacuum (P_V) is applied to the control input named "Con", the PDMS membrane is pulled into a displacement chamber, allowing the fluid to flow between "In" and "Out", as shown in Fig. 1b.

Using valves and etched channels, more complex components can be built, such as switches, pumps, filters, heaters, storage units, detectors and separators [5]. Fig. 2 shows the `AquaFlux` biochip [9] and highlights the nine microfluidic valves, v_1 to v_9, of its pneumatic mixer. The valve-set

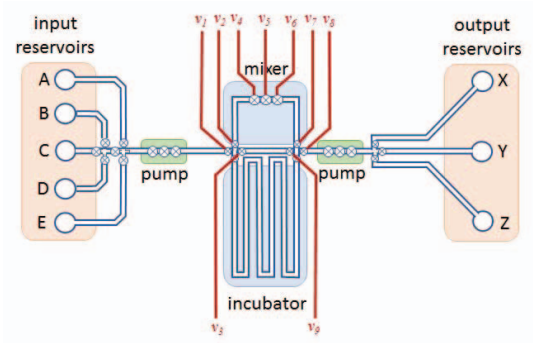


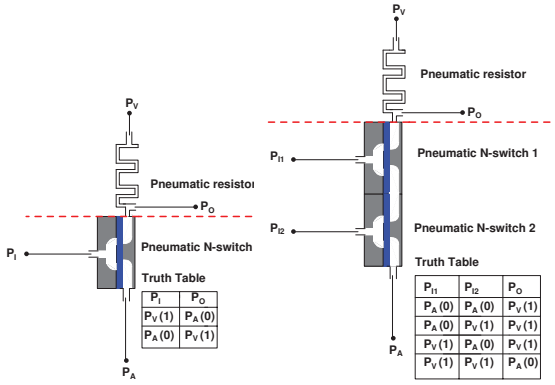
Figure 2. `AquaFlux` biochip architecture [9]

$\{v_1, v_2, v_3\}$ acts as the input switch, $\{v_7, v_8, v_9\}$ as the output switch and $\{v_4, v_5, v_6\}$ as the on-chip pump used to perform the mixing.

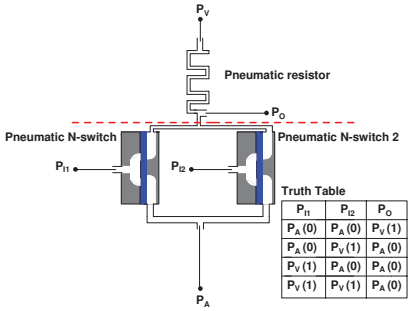
To mix, input and output valves v_1 and v_8 are closed, while valves in the valve-set $\{v_2, v_3, v_7, v_9\}$ are opened. Following this, the mixing operation is initiated by opening and closing valves in the valve-set $\{v_4, v_5, v_6\}$ in a sequence which generates a pumping action [5]. Thus, to control a single mixer, such complex circuitry is necessary to generate the required actuation signals. Additionally, each control signal requires dedicated pressure tubing, bulky off-chip pressure actuators and pumps, thereby limiting them to laboratory environments. As described earlier, on-chip pneumatic control logic solves this problem, by multiplexing the external control signals.

A. On-Chip Pneumatic Control Logic

Traditionally, each of the valves on the biochip is controlled through a dedicated pressure tubing, restricting or permitting the flow of fluid on the flow-layer. To implement the on-chip control circuits, the same valve from Fig. 1 is used to control the flow of gas (air) instead of fluid, which in turn can then be used to manipulate a valve that controls fluid flow.



(a) Pneumatic-NOT gate (b) Pneumatic-NAND gate



(c) Pneumatic-NOR gate

Figure 3. Structure of pneumatic logic gates using normally closed valves and a pneumatic resistor

Coherent with digital logic, we will consider valves to be in either of two possible states, i.e., P_A (logic-0) or P_V (logic-1). There are multiple styles of pneumatic digital logic proposed in prior literature. In [4], vacuum supply corresponds to logic-0, while atmospheric pressure corresponds to logic-1. In [7], the opposite convention is used i.e., vacuum supply corresponds to logic-1, while atmospheric pressure corresponds to logic-0. In this paper, we adhere to the convention in [7].

Figs. 3a-c show the structures of pneumatic-NOT, pneumatic-NAND and pneumatic-NOR gates respectively, constructed using normally closed valves [5] and a pneumatic resistor. The twisted structure between the P_V pin and the output pin is the pneumatic resistor, and its role is similar to the role of resistor in nMOS-resistor based logic in microelectronics. Integrating on-chip control circuits requires multiple steps and might even require to revisit previous steps in the design process, as explained in more detail in the next section.

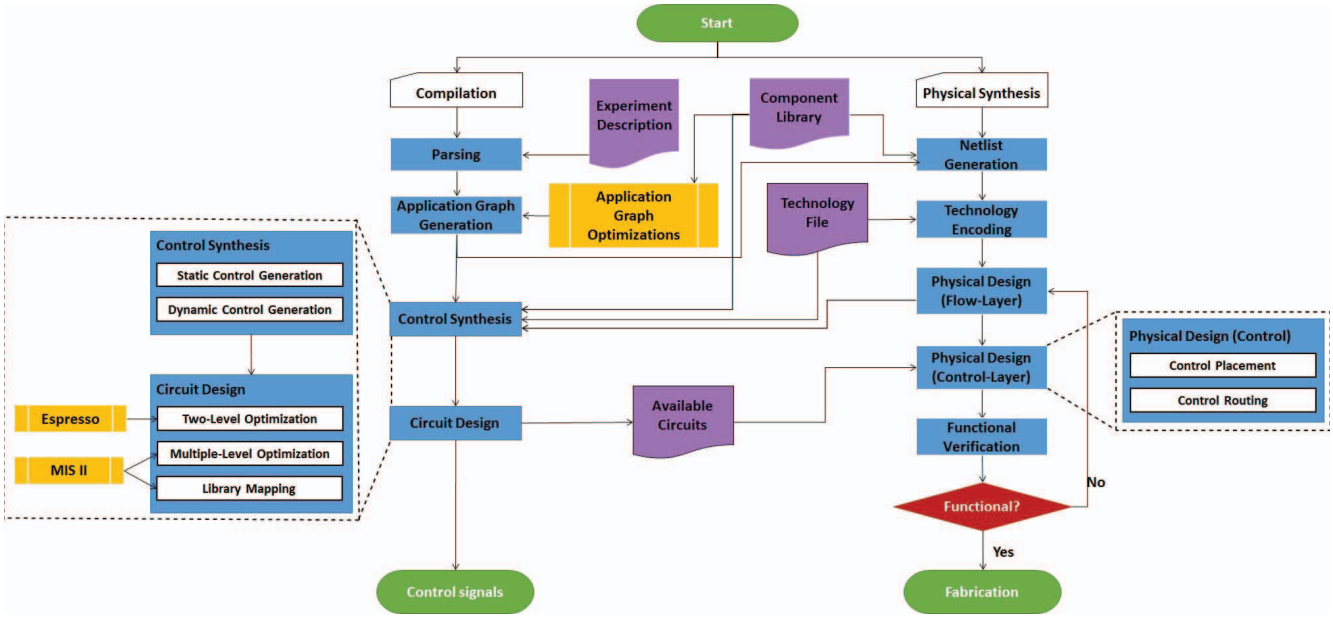


Figure 4. Overview of the design process for a biochip

III. Overview

Fig. 4 shows an overview of the complete design process of a biochip and illustrates where the on-chip control tasks shown, fit into it. The process is logically divided into two branches: Compilation, which determines the control signals for the external hardware driving the biochip, and physical synthesis, which determines the design of the chip to be fabricated. The process requires three data files to be provided by the user: A detailed description about the experiment that is supposed to be executed on the biochip. A component library ζ, providing information about the components that are available for fabrication and the functionality they provide. Furthermore, a technology file τ is required, including information about the material provided for fabrication and its limitations, type of valve to be used, number of layers to be fabricated and other hardware related data. First, the experiment, which can be provided in several forms e.g. as text or a graphical representation, is parsed into an unambiguous form such as Aqua Code [9].

From this, all required operations and the execution sequence can be translated into an application graph α [10]. This application graph is optimized through techniques such as fluid management and adjusted to the available components. For example depending on the available mixing technology (e.g. 1:1 or arbitrary mixing ratios), mixing trees are added to the application graph to suit the components [10]. Knowing all operations, a netlist is generated. Using components from ζ which best suit the operations in α and linking them according to the sequence of the operations results in the first part of the physical synthesis, the netlist N. The technology encoding adjusts N to be compatible with ζ. Major adjustments can originate among others from the type of valve used, or the pumping mechanism (e.g. on-chip vs. off-chip).

The physical design on the flow-layer transforms the abstract data from N into 2D space by placing and routing the components and channels, using dimensions and restrictions provided by ζ and τ [11]. Previous research has offered solutions to flow-layer placement [12]–[17], flow-layer routing [13], [15], [18] and control-layer routing [18]. This physical design step places all components on the chip and therefore also determines the position of the valves, which has a large impact on the design of the control-layer, as valves represent the connection points of these layers on the chip.

Connecting each of the potentially very large number of valves to an external pressure source is a difficult challenge. Solutions towards solving this problem such as pin-count reduction techniques [19] have been proposed and the introduction of on-chip control, which is the main focus of this paper and we discuss in detail next, is an additional step towards solving this issue. The control synthesis step provides data about the valve states throughout the application, which is required by these solutions. Using this data, circuits are designed which provide the correct control for a large number of valves, using only few pressure sources. This can create a wide range of available on-chip circuits, which differ in their efficiency. An efficient circuit is capable of controlling many valves with few pressure sources, however such circuits are complex and therefore space consuming and difficult to route [10].

The physical design for the control-layer is responsible for placing and routing the circuits and therefore also determines which circuits are used, according to the routability, chip area and available pressure sources. The final step is the functional verification, which assures that the given application can be successfully executed on the designed chip. A similar step is

part of each of the previous processes as well, e.g. the physical design on the flow-layer verifies right away if the application can be executed within given deadlines and makes necessary adjustments before continuing to the control synthesis. Some issues are however only detectable at later processes and require to revisit a previous process. It is for example possible that the control-layer cannot be routed successfully with any of the given circuits, and changes to the flow-layer need to be made to solve this problem.

A. Physical Design

Physical design of the flow-layer, as shown Fig. 4 comprises of two subproblems: (1) to position the flow layer components on the rectangular layout; and (2) to route the corresponding flow channels, such that the biochip area is minimized. Subsequently, the control has to be partitioned into on-chip and off-chip components, which has to solved using optimization. Once that is decided, logic synthesis of the on-chip control logic is performed [17].

1) Logic Synthesis: We start from a given biochip architecture, which is specified as a netlist for the flow layer, i.e., the flow layer components and the interconnecting channels. The control synthesis ensures that the synthesized on-chip pneumatic control logic circuit is able to produce all the actuation signals necessary for the proper functionality of all the components in the flow-layer for any biochemical application, i.e., the control is general purpose, given the particular architecture, not application-specific. The first step takes the components in the flow layer netlist as input, and determines component-specific actuation tables, such that all the component operations are achievable. The second step focuses on the control signals needed to assure the correct routing of fluids. Thus, we take the flow layer netlist as input, and we determine the actuation truth tables such that all possible flow paths in the architecture are realizable. The generation of actuation tables which achieve all possible phases of a component in the flow-layer corresponds to the "Static Control Generation" part of "Control Synthesis" shown in Fig. 4. Following this, generating actuation tables such that all possible flow paths between the microfluidic components are realizable, corresponds to the "Dynamic Control Generation".

From the control truth tables, two-level combinational logic minimization and multiple-level combinational logic minimization are done one after another. Finally, the logic minimized circuit is subject to library mapping to obtain a logic level netlist in the target technology standard cell library (of monolithic membrane valve based logic gates). We use existing algorithms to solve these three steps, as shown as subtasks in the "Circuit Design" part of Fig. 4. The next sections explain the subsequent phases of physical synthesis of the control layer. Following that, the physical design of the control-layer, as shown in Fig. 4 comprises of two subproblems: (1) to position the on-chip control logic gates around the rectangular flow layer boundary, shown as "Control Placement"; and (2) to route the required control channels, shown as "Control Routing".

2) Placement: The objective of the first subproblem is to find a suitable placement of all the on-chip control gates, which includes their location, rotation, and layer assignment. To perform this task, the netlists of the on-chip control logic circuits as well as a finished physical design on the flow-layer are required. The physical design on the flow-layer can be performed using a Simulated Annealing metaheuristic [20] and routed using Soukup's algorith, for grid-based routing [10]. Based on these inputs, the algorithm determines the physical positions of the on-chip control logic gates and the control pins such that the distance between connected gates is minimized.

We have used a Simulated Annealing (SA) metaheuristic [20], which is implemented as a loop that iteratively improves the current solution by exploring its offspring solutions (also called neighboring solutions). These replacement candidates are then evaluated based on a parameter called temperature, which decreases with time, and their quality evaluated using a objective function. The offspring of a solution is generated based on a set of defined permutation possibilities. The offspring is found by executing one of five possible operations on the current solution. These operations can be divided into three different groups which are discussed below.

Move and Slide. The "Move" operation relocates the component to a completely new position, while the "Slide" operation is restricted to moving a component either horizontally or vertically. Both of the operations are restricted such that the positioning is gridded. This is a hard constraint such that all permutations are feasible solutions which are physically implementable.

Switch. The "Switch" operation is intended to explore solutions where components exchange their positions. This is done by selecting two components at random and interchanging their positions. This is a hard constraint such that all permutations are feasible solutions which are physically implementable.

Invert. Since both the control and the flow layer are used for the control circuits when using normally closed valves, components can be places in a normal or inverted state. In the inverted state, channels that before occupied the flow-layer, are now located in the control-layer and vice versa. To connect channels from different layers, layer changes are required which are represented by yellow dots in Fig. 5a. Inverting the layer assignment of a component can have a significant impact on a circuits routing and the amount of layer changes required as visible in Fig. 5b. The *Invert* operation selects a random component and inverts its layer assignment. This operation will never result in a violation of overlapping components since the specific area required by the component is not changed.

Rotate. The optimal rotation should minimize the length of inter component connections. The rotation of components has been restricted to four different rotations. Each of the components are designed in a left to right manner meaning that the inputs are primarily on the left and output on the right. This is used to guide the rotation of the components.

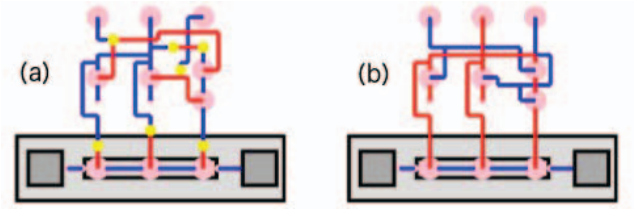


Figure 5. On-chip control circuit connected to three control pins in the normal and inverted state. Red and blue lines represent channels on the flow- and control-layer respectively, yellow dots mark layer-changes.

A specific rotation is subject to the hard constraint that the component does not overlap with other components. Again this ensures that the solution is valid.

Objective Function. Several objective functions have over the years been suggested for placement of components. To maintain the focus of the current paper, we do not discuss all the available cost functions in the literature. We have used the squared two dimensional Manhattan Distance, as a partial evaluation objective function. Using Manhattan distance provides an adequate level of accuracy without being too computationally heavy. The suggested objective function furthermore exists both in a normal and squared version. The squared version was selected since it penalizes long distances.

$$L_M = \sum_{c_{uv}} \left(|u_x - v_x| + |u_y - v_y|\right)^2 \quad (1)$$

A second objective function is applied to the Invert operation. The layer of each of the ports in the inverted component is compared to the closest port to which it is connected. The cost of an inverted state is equal to the number of ports pairs which are not on the same layer. The closest port is selected since it is most likely to be connected to that specific port.

Delta Evaluation. The number of inter-component connections considered tends to be quite large. This means that calculating the Manhattan distances for all inter-component connections is infeasible. It is furthermore not necessary to recalculate a distance at each iteration. Utilizing a delta evaluation approach can reduce the number of calculations required for each iteration. To calculate the difference between the parent and the offspring it is enough to look at the connections related components that were changed.

3) Routing: The routing problem takes the (1) flow architecture, (2) the placed on-chip control circuit, and (3) placed control pins as input, and determines physical routing of the control channels, such that (1) all channels are routed correctly, (2) the design rules are respected and (3) the length of control channels is minimized. We used a modified three-dimensional version of Lee's path finding algorithm [21] for routing. Similar to VLSI, the on-chip control logic routing is composed of global routing, followed by local routing. Since the net ordering is of high importance, we used the difference between the Manhattan distance of the ports and the defined maximum length of the net which considers both the net-length and timing-criticality [10].

In global routing phase, the entire biochip area is partitioned into tiles, which are used to find a preferred path, which is used in the local routing phase. Once the global routing phase is completed, the local routing phase determines the exact path connecting a number of targets while preferably avoiding congested areas. More details of the routing implementation are available in [10].

IV. Experimental Evaluation

We have used our proposed methodology to synthesize six biochips, AquaFlux, Urbanski, PCR1–3 and EA1. For details about the functionality of these biochips the reader is directed to [10]. We make the following design rules and assumptions in our experiments: (1) Flow channels and control channels require a minimum of 75 μm width and a minimum of 75 μm distance to nearest channel; (2) Layer change requires at least 225 μm from its center to nearest channel; (3) Vents require at least 75 μm from its center to nearest channel; (4) The size of a normally closed valve used in logic gates is 450 μm by 450 μm; and (5) Control channels cannot be parallel with other control channels, i.e., above and below. A control channel can however be parallel with itself.

We assume that the on-chip vs. off-chip control partitioning has been done manually. For all the experiments in this sections we have decided that all of the control should be placed on-chip. Note that this may not be the optimal solution to the on/off-chip control partitioning problem, since it may result in large on-chip control circuits. We will address the on/off-chip control trade-off problem in our future work. We have used the Logic Synthesis approach from Section III-A1 to generate general purpose on-chip control circuits.

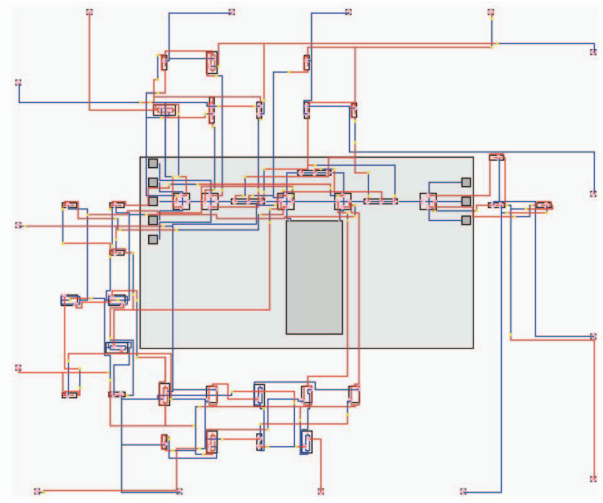
The placement of the flow and control layers, considering

Table I
Results for on-chip control placement

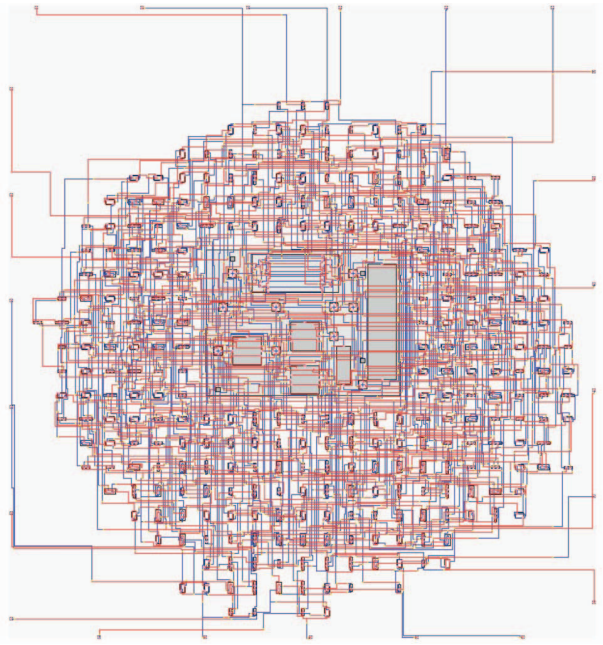
Biochip	Full Area	Factor	Min. No. of Layer Changes	Min. Length of Control Lines
AquaFlux	79,722	4.56	18	4,619
Urbanski	107,880	5.50	25	5,956
PCR1	111,240	6.90	34	6,408
PCR2	167,281	8.85	88	16,635
PCR3	208,690	10.03	125	24,826
EA1	401,380	13.38	264	79,181

Table II
Results for on-chip control routing

Biochip	Min. No. of of Layer Changes	Actual No. of Layer Changes	Min. Length of Control Lines	Actual Length of Control Lines
AquaFlux	18	107	4,619	5,831
Urbanski	25	139	5,956	7,828
PCR1	34	159	6,408	8,056
PCR2	88	534	16,635	21,315
PCR3	125	810	24,826	32,413
EA1	264	2,322	79,181	90,956



(a) AquaFlux biochip



(b) EA1 biochip

Figure 6. Biochip layouts after applying our on-chip control physical design methodology

the synthesized on-chip control circuits, is done using the Simulated Annealing algorithm presented in Section III-A2. The obtained placement results are summarized in Table I. The table shows the area of the full architecture, the number of layer changes and length of control lines. The results of the routing of the flow and control layers, considering on-chip control, are summarized in Table II, which shows the number of layer changes and length of control lines.

Fig. 6 shows the AquaFlux and EA1 biochips after applying our on-chip control physical design methodology. In the figures, the dark gray regions represent the flow processor and the area surrounding it is the on-chip control logic. Note that for these examples all the control has been included on-chip, and the control is general purpose, i.e., it can handle the control for any applications that would run on these biochips.

V. Conclusions

In this paper we have addressed flow-based mVLSI biochips that use normally-closed valves, which can be used to control both the flow of fluid, but also the pressure used to drive other valves, forming this pneumatic on-chip control circuits. We have presented an overall methodology for the compilation and physical design of mVLSI biochips, and discussed how on-chip control fits into such a methodology. We have discussed the problem of partitioning the control between on-chip and off-chip. We have proposed a technique for the logic synthesis of control circuits, and extended existing placement and routing algorithms for the physical design of flow and control layers to take into account the on-chip control circuits. The proposed approaches have been evaluated by synthesizing several biochip benchmarks.

References

[1] D. Mark et al, "Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications," *CSR*, vol. 39, pp. 1153–1182, 2010.

[2] I. E. Araci et al, "mVLSI with integrated micromechanical valves," in *Lab on Chip*, vol. 12, pp. 1463–1475, 2012.

[3] J. W. Hong et al, "Integrated nanoliter systems," *Nature Biotechnology*, vol. 21, pp. 1179–1183, 2003.

[4] E. C. Jensen et al, "Micropneumatic digital logic structures for integrated microdevice computation and control," *IEEE Journal of Micromechanical Systems*, vol. 16, no. 6, pp. 1378–1385, 2007.

[5] W. H. Grover et al, *"Monolithic Membrane Valves and Pumps" chapter in Lab-on-Chip Technology*. Caister Academic Press, 2009.

[6] T. Nguyen et al, "Microfluidic finite state machine for autonomous control of integrated fluid networks," *Proc. of Micro Total Analysis Systems*, pp. 741–743, 2011.

[7] P. N. Duncan et al, "Pneumatic oscillator circuits for timing and control of integrated microfluidics," *PNAS*, vol. 110, no. 45, pp. 18104–18109, 2013.

[8] T. Thorsen et al, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002.

[9] A. M. A. Amin et al, "Aquacore: A programmable architecture for microfluidics," in *ISCA*, pp. 254–265, 2007.

[10] P. Pop et al, *Microfluidic Very Large Scale Integration (VLSI): Modeling, Simulation, Testing, Compilation and Physical Synthesis*. Springer International Publishing, 2016.

[11] W. H. Minhass, P. Pop, and J. Madsen, "System-level modeling and synthesis of flow-based microfluidic biochips," in *CASES*, 2011.

[12] J. McDaniel et al, "Simulated Annealing-based Placement for Microfluidic Large Scale Integration (mLSI) Chips," in *VLSI-SoC*, pp. 213–218, 2014.

[13] W. H. Minhass et al, "Architectural synthesis of flow-based microfluidic large-scale integration biochips," in *CASES*, pp. 181–190, 2012.

[14] K.-H. Tseng et al, "A top-down synthesis methodology for flow-based microfluidic biochips considering valve-switching minimization," in *ISPD*, pp. 123–129, ACM, 2013.

[15] C.-X. Lin et al, "An efficient bi-criteria flow channel routing algorithm for flow-based microfluidic biochips," in *DAC*, pp. 1–6, ACM, 2014.

[16] M. Raagard, "Placement algorithm for Flow-based Microfluidic Biochips," 2014.

[17] M. S. Hørslev-Petersen, "Computer-aided design for the physical synthesis of biochip control logic," Master's thesis, DTU, 2015.

[18] K. Hu et al, "Control-layer optimization for flow-based mvlsi microfluidic biochips," in *CASES*, p. 16, ACM, 2014.

[19] P. P. A. Schneider and J. Madsen, "A pin-count reduction algorithm for flow-based microfluidic biochips," in *DTIP*, 2016.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," 1983.

[21] N. A. Sherwani, "Global routing," in *Algorithms for VLSI Physical Design Automaton*, Kluwer Academic Publishers, 2002.