DTIP 16-18 April 2013, Barcelona, Spain

Modeling and Simulation Framework for Flow-Based Microfluidic Biochips

Morten Foged Schmidt, Wajid Hassan Minhass, Paul Pop, Jan Madsen
DTU Compute
Technical University of Denmark
DK-2800 Kgs. Lyngby, Denmark

***Abstract* - Microfluidic biochips are replacing the conventional biochemical analyzers and are able to integrate the necessary functions for biochemical analysis on-chip. In this paper we are interested in flow-based biochips, in which the fluidic flow is manipulated using integrated microvalves. By combining several microvalves, more complex units, such as micropumps, switches, mixers, and multiplexers, can be built. Such biochips are becoming increasingly complex, with thousands of components, but are still designed manually using a bottom-up full-custom design approach, which is extremely labor intensive and error prone. In this paper, we present an Integrated Development Environment (IDE), which addresses (i) schematic capture of the biochip architecture and biochemical application, (ii) logic simulation of an application running on a biochip, and is able to integrate the high level synthesis tasks we have developed for the top-down design of flow-based biochips. We show how the IDE can be used to design biochips for several applications.**

***Index Terms* - CAD, Microfluidics, Biochips, Simulation, IDE**.

I. INTRODUCTION

Biochips integrate different biochemical analysis functionalities (e.g., dispensers, filters, mixers) on-chip, miniaturizing the macroscopic chemical and biological processes to a sub-millimeter scale [1]. Microfluidic biochips can readily facilitate clinical diagnostics, enzymatic and proteomic analysis, cancer and stem cell research, and automated drug discovery [2].

There are several types of microfluidic biochip platforms, each having its own advantages and limitations [7]. In this paper, we focus on the flow-based biochips in which the microfluidic channel circuitry on the chip is equipped with chip-integrated microvalves that are used to manipulate the on-chip fluid flow [1]. By combining several microvalves, more complex units like mixers, micropumps, multiplexers etc. can be constructed, with hundreds of units being accommodated on one single chip. This approach is called microfluidic Large Scale Integration (mLSI) [1].

A. Related Work

Current tools and manual approaches do not scale with the increasing complexity of biochips (commercial biochips are available which use more than 25,000 valves and about a million features to run 9,216 polymerase chain reactions in parallel [2]). Designers use drawing tools such as AutoCAD [5] to draw the masks used in the soft-lithography fabrication process. A template file is provided by the microfluidic foundry showing the designers how to manually draw the biochip design. The microfluidic foundry also provides a set of design rules, which the designer has to follow [6]. Manual approaches are used to verify the correctness of the designs, and the tools do not have the ability to simulate execution of biochemical applications on biochips before physically constructing them [4]. The design of a biochip with 918 valves [11] can take up to one year of post-doctoral researcher time [12]. Therefore, new top-down Computer-Aided Design (CAD) tools are required, which can offer the same level of support as the one taken for granted in the semiconductor industry [3].

Although biochips are becoming more complex, CAD tools for these chips are still in their infancy. Most CAD research has been focused on device-level physical modeling of components. Researchers have proposed significant work on the top-down synthesis methodologies for digital biochips [9], which manipulate the liquid as discrete droplets on an electrode array. However, these techniques are not applicable to flow-based biochips. Recently, we have proposed High-Level Synthesis (HLS) tools [10] for flow-based biochips, which starting from a biochemical application model and a library of components, automatically synthesize a biochip architecture and the control required to run the application.

B. Contribution

In this paper we propose an Integrated Development Environment (IDE) for flow-based microfluidic biochips. The IDE offers modeling, simulation and visualization capabilities, and is used as a framework which integrates our HLS tools. The IDE offers (i) schematic capture of the biochip architecture and the biochemical application, (ii) logic simulation of an application running on a biochip, and uses XML files to integrate various synthesis tools. To the best of our knowledge, it is the first time such an IDE has been proposed for flow-based biochips.

II. DESIGN FLOW

The proposed design flow is presented Fig. 1. The main design tasks are briefly outlined below. This paper focuses on the *Architectural Synthesis* and *Simulation and Visualization* tasks, highlighted in blue in Fig. 1. The goal is to output low-level models (AutoCAD mask drawings), which can be directly used as input to the fabrication process.

©EDA Publishing/DTIP 2013 ISBN:

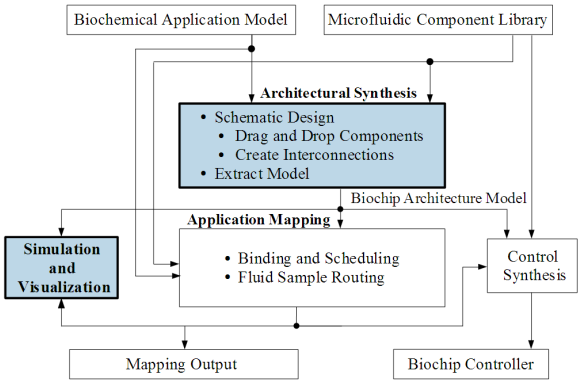


Fig. 1. Design Flow

Modeling

We propose models for the biochemical applications, biochip architecture and components. The models are presented in Section III. We use these models as internal data structures in our IDE and HLS tools.

Architectural Synthesis

During this task, the architecture of the biochip is decided. Components are selected from a library of available components (Mixers, Filters, Heaters, Detectors, etc.) and interconnect using flow channels. The architecture synthesis task is composed of the following steps: allocation of components from a library, placement of components in a given area, routing of fluidic channels and control channels and finally mask generation for fabrication. Our IDE offers two options: (i) a fully automatic architectural synthesis, starting from a biochemical application model, which uses our proposed HLS tools to synthesize an architecture, going through all the steps mentioned earlier; and (ii) a schematic capture editor, where the designer manually builds a schematic of the biochip architecture (see Fig. 2 for an example).

Application Mapping

Once a biochip architecture and a biochemical application is available, the biochemical application has to be mapped onto the biochip. The steps involved are; binding and scheduling of biochemical operations onto the allocated components and performing fluidic routing, while satisfying any resource and dependency constraints. The next step is control synthesis.

Control Synthesis

During the control synthesis task, the exact sequence of valve activations is extracted from the schedule generated in the previous task. This activation sequence can be fed to a biochip controller, in order run the biochemical application.

Simulation and Visualization

An important tasks during the design flow is the logic simulation task. During logic simulation, the designer simulates the execution of a biochemical application on a biochip architecture in order to validate the design before the biochip is fabricated. Thus, the overall design can be verified and validated before going though time-consuming and expensive fabrication steps. Hence, logic simulation is crucial for speeding up the design cycle of the biochips. In addition, our proposed IDE has a Graphical User Interface (GUI), which offers visualization during all tasks of the design flow. Simulation and visualizations enhance comprehensibility and thus speed up the design cycle.

We have designed an Integrated Development Environment (IDE), which incorporates the above tasks see Fig. 10.

III. System Model

This section presents the models used in the IDE and HLS tools.

A. Biochip Architecture

Fig. 3b shows the schematic view of a flow-based biochip with four input ports and three output ports, a mixer, a filter and a detector. Fig. 3c shows the functional view of the same chip. The biochip is manufactured using multilayer soft lithography [1]. A cheap, rubber-like elastomer (polydimethylsiloxane, PDMS) with good biocompatibility and optical transparency is used as the fabrication substrate [1]. Physically, the biochip can have multiple layers, but the layers are logically divided into two types: *flow layer* (depicted in blue) and the *control layer* (depicted in red) [1].

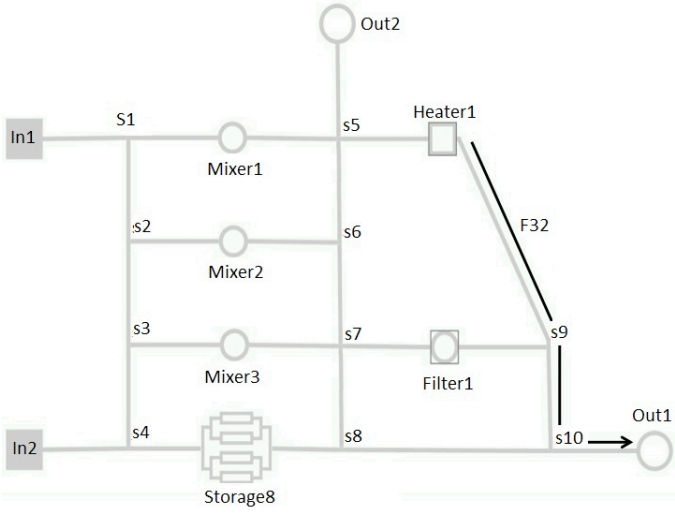
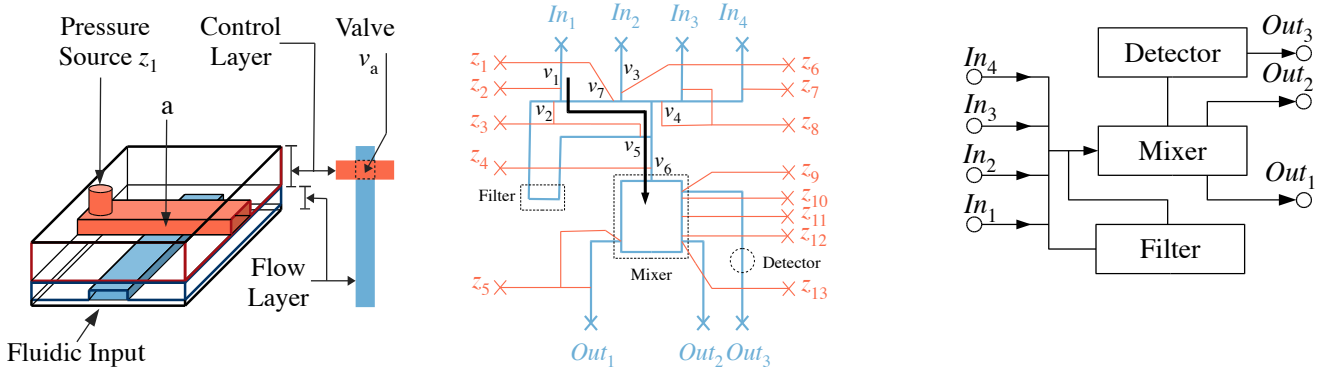


Fig. 2. Architecture Schematic



(a) Microfluidic Valve (b) Schematic View (c) Biochip: Functional View

Fig. 3. Flow-based Biochip Architecture

The liquid in the flow layer is manipulated using the control layer. The basic building block of such a biochip is a valve (see Fig. 3a), which is used to manipulate the fluid in the flow layer as the valves restrict/permit the fluid flow. The control layer (red) is connected to an external air pressure source z_1. The flow layer (blue) is connected to a fluid reservoir through a pump that generates the fluid flow. When the pressure source is not active, the fluid is permitted to flow freely (open valve). When the pressure source is activated, high pressure causes the elastic control layer to pinch the underlying flow layer (point a in Fig. 3a) blocking the fluid flow at point a (closed valve). Because of their small size ($100 \times 100\ \mu m^2$) a biochip can accommodate thousands of valves. By combining these valves, more complex units, such as switches, multiplexers, micropumps, mixers, can be constructed [1].

B. *Biochip Architecture Model*

The biochip architecture is modeled using a topology graph model [10], which captures the components and their interconnections.

The basic component is the microvalve and several of these can be combined to create more complex components, such as switches, mixers, storage etc. The switch component directs the fluidic flow in the flow layer. The switch shown in Fig. 4., can direct an incoming flow in three new directions. The switch has four microvalves ($v1$, $v2$, $v3$, $v4$), which can manipulate the flow direction, depending on the combination of closed valves. The microvalves are controlled by the pressure sources ($z1$, $z2$, $z3$, $z4$). For example, if a flow from the top to the right flow channel is performed, $v1$ and $v4$ are opened and $v2$ and $v3$ are closed.

A microfluidic mixer is graphically presented in Fig. 5. The mixer has nine microvalves ($v1$-$v9$) controlling the component and its operations is modeled as five phases. Table I shows the phases of the mixer, for each phase, the valves have to be opened (0) or closed (1). If the top channel needs to be filled with a sample, the mixer should be in phase ip1, here $v3$ and $v9$ are closed and the rest of the valves are open. A new phase should be entered when the bottom channel needs to be filled. The mixer should then change phase to ip2. When the mixer has a sample in each channel (top and bottom), the mixer phase should change to mix, which transforms the mixer into a circular architecture by closing $v1$ and $v8$. The mix is performed using the on-chip pump ($v4$, $v5$, $v6$), the pump works by opening and closing the valves, $v4$, $v5$, $v6$, with a fixed frequency that pushes the fluid around until it has been mixed. When the mixed samples are needed elsewhere the mixer is emptied by changing phase to op1 or op2 [13].

Our IDE takes as an input a library of components described in an XML file. The IDE loads the XML content and thereby is capable of visualizing biochip architectures.

C. *Biochemical Application Model*

The biochemical application is modeled using a sequencing graph [3], which can be obtained from a biochemical protocol description language such as BioCoder [14], by compilation. The graph $G(O,E)$ is directed, acyclic and polar (i.e., there is a *source* vertex that has no predecessors and a *sink* vertex that has no successors).

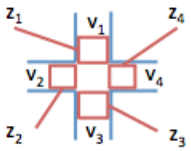


Fig. 4. Graphical Representation of a Switch

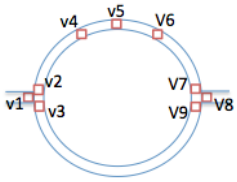


Fig. 5. Graphical Representation of a Mixer

TABLE I. MIXER PHASES

Phase	v1	v2	v3	v4	v5	v6	v7	v8	v9
ip1	0	0	1	0	0	0	0	0	1
ip2	0	1	0	0	0	0	1	0	0
mix	1	0	0	Mix	Mix	Mix	0	1	0
op1	0	0	1	0	0	0	0	0	1
op2	0	1	0	0	0	0	1	0	0

©EDA Publishing/DTIP 2013 ISBN:

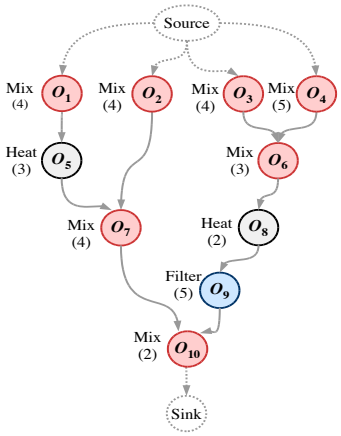


Fig. 6. Biochemical Application Model

Each vertex $O_i \in O$ represents an operation that can be bound to a component $M_j \in \mathcal{M}$ using a binding function $\mathcal{B} : O \rightarrow \mathcal{M}$. Each vertex has an associated weight $C_i^{M_j}$, which denotes the execution time required for O_i to be completed on M_j. Fig. 6 shows an example of a biochemical application model which has seven mixing operations (O_1–O_4, O_6, O_7, O_{10}), one filtration operation (O_9) and two heating operations (O_5, O_8). The execution times for the operations are also given (the parameter below the operation type).

IV. Integrated Development Environment

A. Architectural Synthesis

Our IDE allows the design of biochemical applications and biochip architecture, by use of a schematic editor. Using the schematic editor, a designer is able to visually construct a schematic architecture, as shown in Fig. 2, by dragging and dropping components to a drawing board in the IDE. The IDE can save schematics to a file and load architecture schematics, either manually designed or automatically produced by our HLS tools. A designer can quickly redesign the biochip by changing the components, their parameters and the way they are interconnected.

Note that fluid samples do not occupy the full channel length between biochip components; they occupy a fixed length of the flow channel. We assume that the fluids are a multiple of a fluid unit decided by the designer. The required fluid volumes are produced by a metering unit, which can be placed at the input ports [10]. External fluidic input ports are connected to a pump and a filler oil reservoir that makes samples move within the biochip [8]. To make a fluid sample move from, e.g., *Mixer*1 to *Out*1 (see Fig. 2); Mixer1 needs to be connected to an input source e.g. *In*1. The channel connection is necessary because a pressure is needed to drive the fluid flow. The destination *Out*1 needs to be connected to an output port or sink to absorb the pressure generated from *In*1. Microvalves available on the biochip will establish a connection between the pressure source and sink (valves placed in *s*1 and *s*5). When a connection is established, a pumping action is created where filler oil flows in the direction of *Out*2. This action makes the fluidic sample move from *Mixer*1 to *Out*2. The pumping action stops when the fluidic sample has reached its destination. Our schematic editor automatically takes care of these aspects.

B. Generation of FlowPaths

Our IDE integrates several CAD tools, which implement the various designs tasks. The integration of these tools is not straightforward.

Hence, we have proposed an algorithm for extracting the flowpaths from an architecture schematic. The algorithm is presented in Fig. 7. The algorithm recursively produce all flowpaths associated with a component $\mathcal{M}1$. In line 2, $\mathcal{M}1$ is added to the flowpath $\mathcal{F}1$. In line 4 a connected component is found, $\mathcal{M}2$. In line 5 a check is performed, if $\mathcal{M}2$ is a linking component such as a switch component where no operation (mixing, heating, filtering etc.) can be performed, and has multiple flow directions, the flowpath $\mathcal{F}1$ is cloned and the algorithm recursively calls itself to calculate all flows. The process stops when the component capable of performing an operation (mixing, heating, filtering etc.) or an output port/sink is found. When all available flows have been found, the microvalve combinations are extracted and added to the flowpaths. Finally, all routing constraints are added to the flowpaths.

The extracted flowpaths are placed in a flowpath table as shown in Fig. 8. The flowpath example shown in Fig. 2 (Black Line) is named *F*32 (see Fig. 8), which is unique for this particular flow. In Fig. 8, "FlowPath" is the list $\mathcal{G}$ components that the fluidic sample passes *Heater*1-*S*9-*S*10-*Out*1. The "SinkPath" is all pressurized channels and components responsible for the movement of a fluid. "Time" states the time it takes for a sample to move from the first component to the last in the "FlowPath" component list, in this case it takes 2 seconds (0,5s (*Heater*1) + 0,6s (*Heater*1-*s9*) + 0,2 (*s*9-*s*10) + 0,2s (*s*10-*Out*1) + 0,5s (*Out1*) = 2 seconds), the calculation is based on two parameters. The channel length and the speed the pressure sources can produce, called flow rate. A typical flow rate could be 10 mm/s. The "Open Valves" and "Closed Valves" needed to direct the flow are also given. The "Routing Constrains" identify the flowpathsets that cannot be executed at the same time as they share the flow channels.

FindPaths $\mathcal{G}(\mathcal{F}1, \mathcal{M}1)$

1. Initialize $\mathcal{M}2$, $\mathcal{F}2$
2. AddComponent($\mathcal{F}1$, $\mathcal{M}1$)
3. **for all** <All connection channels on the biochip> **do**
4. $\mathcal{M}2$ = ConnectedComponent(*Con*, $\mathcal{M}1$)
5. **if** <$\mathcal{M}2$ is linking component> **then**
6. $\mathcal{F}2$ = CloneFlowPath($\mathcal{F}1$)
7. UpdateExecutionTime($\mathcal{F}2$, $\mathcal{M}2$)
8. FindPaths($\mathcal{F}2$, $\mathcal{M}2$)
9. **else**
10. AddComponent($\mathcal{F}1$, $\mathcal{M}1$)
11. **end if**
12. **end for**

Fig. 7. Flowpath algorithm for flow-based biochips

Flow Path Sets

Name	Flow Path	Sink Path	Closed Valves	Open Valves	Time	Constrains
F32	Heater1 S9 S10 Out1	Heater1 S9 S10 Out1	z57 z61	z56 z58 z60 z63	2	F8-1 F8-2 F10-1 F10-2 F13-1 F13-2 F
F31	Heater1 S9 S10 S8 S7 Filter1	Heater1 S9 S10 S8 S7 Filter1	z57 z63 z53 z48 z49	z56 z58 z60 z61 z52 z55 z50 z51	3.4	F8-1 F8-2 F9-1-1 F9-1-2 F9-1-3 F9-1
F30	Heater1 S9 S10 S8 S7 S6 S5 Out2	Heater1 S9 S10 S8 S7 S6 S5 Out2	z57 z63 z53 z49 z51 z45 z41 z43	z56 z58 z60 z61 z52 z55 z48 z50 z44 z46 z40 z42	4.2	F7-1 F7-2 F8-1 F8-2 F9-1-1 F9-1-2 F9
F29-8	Heater1 S9 S10 S8 Storage8	Heater1 S9 S10 S8 Storage8	z57 z63 z52 z64 z66 z68	z56 z58 z60 z61 z53 z55 z65 z67 z69	2.9	F8-1 F8-2 F9-1-1 F9-1-2 F9-1-3 F9-1
F29-7	Heater1 S9 S10 S8 Storage8	Heater1 S9 S10 S8 Storage8	z57 z63 z52 z64 z66 z69	z56 z58 z60 z61 z53 z55 z65 z67 z68	2.9	F8-1 F8-2 F9-1-1 F9-1-2 F9-1-3 F9-1

Fig. 8. Snapshot of FlowPathSets

C. Simulation and Visualization

To illustrate flow-based biochip simulation, let us consider an example where a biochemical application modeled as a sequence graph, is executed on a biochip. The logic simulation takes as input a biochemical application model, a schematic model of the biochip architecture (which can be designed manually or automatically synthesized by our HLS tools), the binding of application operations to the biochip components and the schedule of operations (obtained by the application mapping task). The example uses the biochip architecture shown in Fig. 2 and the biochemical application graph shown in Fig. 6. The application graph shows that four inputs from fluidic sources are needed. There are several operations that should be performed (mix, heat, filter) and with individual specified execution times, e.g., operation $O1$ performs a mix operation in 4 seconds. The application finishes when $O10$ has been executed and the fluidic sample in $O10$ has been moved to a sink.

The schedule for this example is depicted in Fig. 9. The schedule is represented as a Gantt chart, where the length of the rectangles captures the execution time of an operation. All blue rectangles are fluidic sample movement between components. The red, green and black rectangles are operations executed in their relevant components, shown to the left (Mixer1, Mixer2, Mixer3, Heater and Filter). The main purpose of the logic simulation is to verify and validate the chip design before the costly and time-consuming fabrication phase. For example, by using the logic simulator, it is possible to go step-by-step through the schedule and check that everything works as expected (see Fig. 10). One example could be the mix operation; a mix operation needs two fluids to operate correctly. It is difficult to see from the schedule if there are two fluids in the mixers when the operations are executed. Another critical area is the difficulty of verifying whether a fluidic sample movement will collide with another fluidic flow. The IDE clearly shows if two fluids are colliding see Fig. 11. The IDE provides a log with collected information from the simulation. This allows the designer to validate the state of the fluidic sample and the microvalves state, see Fig. 12 and 13. Biochips are limited in many ways [6]; one is the number of pressure sources. The microvalve control log data makes it possible to optimize the use of pressure sources. One example could be if two microvalves are closed and opened at the same time through an execution, they can utilize the same pressure source and thereby minimize the use of pressure sources.

The microvalve information can also be used for the final Control Synthesis design task [10] before loading the control data into a real biochip controller.

IV. Evaluation

We have evaluated our proposed modeling, simulation and visualization framework by using several case studies. The application graphs were drawn using the editor of the IDE. The schematic model of the applications have been drawn manually for each case study, as well as generated automatically by the integrated HLS tools. Table II shows the details of the case studies. Case 1 in Table II presents the mixing stage of a Polymerase Chain Reaction (PCR) application. All cases have been executed and a video recording has been made. The recordings are available on the IDE home page [15], located under the evaluation webpages. Fig. 10 shows a screenshot of our proposed IDE.

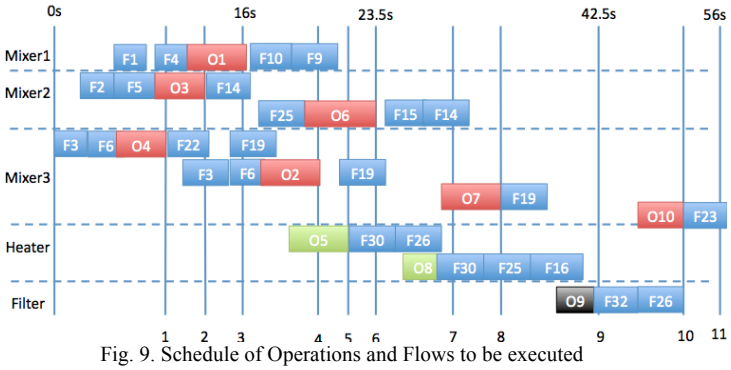


Fig. 9. Schedule of Operations and Flows to be executed

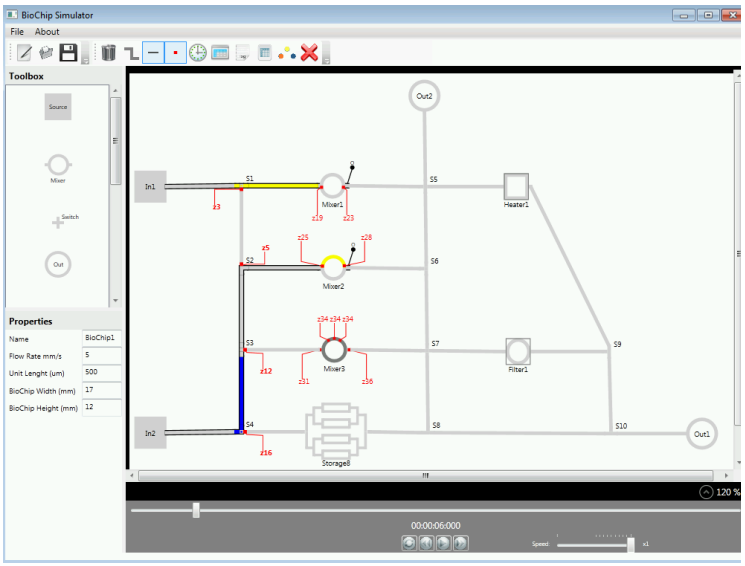


Fig. 10. IDE for Modeling, Simulation, Visualization and Design

 ISBN:

IV. Conclusions and Future Work

In this paper we have presented a framework for modeling and simulation of flow-based microfluidic biochips. The framework tackles the increasing complexity of biochips, by offering support for editing, simulation and visualization and by automating the design tasks that are currently performed manually. An IDE has been proposed to support the design process and enable designers to test their designs before physically building them. Having evaluated the framework using case studies we believe that this new top-down design approach is a more efficient and effective way to design flow-based microfluidic biochips. Our goal is to integrate into the IDE a tool for mask generation, and thus output AutoCAD files, which can be used as an input to the fabrication process. Another aspect to be addressed is the compilation of protocol description languages such as BioCoder to our graph-based biochemical application model.

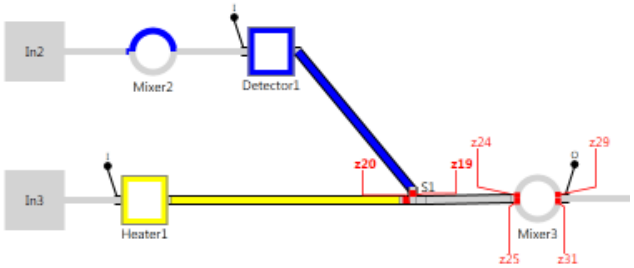


Fig. 11. Fluidic Sample Collision

Time	Fluid
00:00:34:100	Heat(Filter(Mix(F1 + F1)))
00:00:34:000	Heat(Filter(Mix(F1 + F1)))
00:00:32:100	Filter(Mix(Detect(Mix(F2 + F2)) + Heat(F3)))
00:00:32:000	Filter(Mix(Detect(Mix(F2 + F2)) + Heat(F3)))
00:00:31:900	Filter(Mix(Detect(Mix(F2 + F2)) + Heat(F3)))

Fig. 12. Fluidic Log

Time	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12
00:00:08:100	0	0	1	0	0	0	0	0	1	1	0	0
00:00:08:200	0	0	1	0	0	0	0	0	1	1	0	0
00:00:08:300	x	x	x	x	x	x	x	x	x	1	0	0

Fig. 13. Control Layer Data (Open (0), Closed (1) and Free (x))

TABLE II. Case Studies

Case	Type	Components	Operations	FlowPaths
1	Real Life	23	7	42
2	Synthetic	16	10	23
3	Synthetic	36	20	66
4	Synthetic	52	30	92

References

[1] G T. Thorsen, S. J. Maerki, and S. R. Quake, "Microfluidic large-scale integration," Science, vol. 298, no. 5593, pp. 580–584, October 2002.

[2] J. M. Perkel, "Microfluidics - bringing new things to life science," Science, November 2008.

[3] K. Chakrabarty and J. Zeng, "Design automation for microfluidics-based biochips". Journal on Emerging Technologies in Computing Systems, 1(3): 186–223, 2005.

[4] "Test Your Device". Stanford Microfluidics Foundry. Retrieved from http://www.stanford.edu/group/foundry/Testing%20Your%20Device.html

[5] "Getting started AUTOCAD". Stanford Microfluidic Foundry. http://www.stanford.edu/group/foundry/Getting_started_AUTOCAD.html

[6] "Basic Design Rules". Stanford Microfluidic Foundry. http://www.stanford.edu/group/foundry/Basic%20Design%20Rules.html

[7] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, "Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications." Chem. Soc. Rev., 39:1153–1182, 2010.

[8] J. P. Urbanski, W. Thies, C. Rhodes, S. Amarasinghe, and T. Thorsen, "Digital microfluidics using soft lithography". Lab Chip, 6:96–104, 2006.

[9] K. Chakrabarty and T. Xu, "Digital Microfluidic Biochips: Design Automation and Optimization". CRC Press, 2010.

[10] W. H. Minhass, "System-Level Modeling and Synthesis Techniques for Flow-Based Microfluidic Very Large Scale Integration Biochips". Ph.D. Thesis, Technical University of Denmark, 2013

[11] L. M. Fidalgo, S. J. Maerk. "A software-programmable microfluidic device for automated biology". Lab On A Chip, M: 1612-1619, 2011

[12] J. McDaniel, A. Baez, B. Crites, A. Tammewar, P. Brisk "Design and Verification Tools for Continuous Fluid Flow-Based Microfluidic Devices". Asia and South Pacific Design Automation Conference, 219-224, 2013

[13] H. Chou, M. A. Unger, S. R. Quake "A Microfabricated Rotary Pump". Journal of Biochemical Microdevices, 3(4), 323-330, 2001

[14] V Ananthanarayanan, W. Thies "Biocoder: A programming language for standardizing and automating biology protocols". J Biol Eng. 2010 Nov 8;4:13.

[15] M. F. Schmidt "Biochip Simulator Website". https://sites.google.com/site/biochipsimulator/