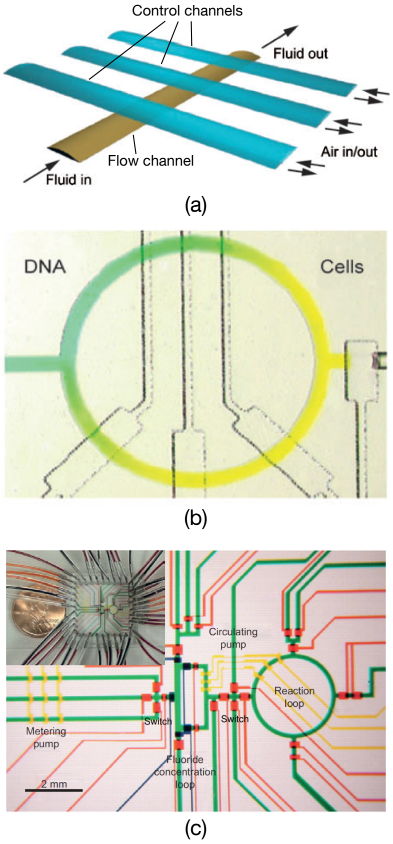
Synthesizing Flow-based Microfluidic Biochips

—Bridging Microfluidics and Biochemical Applications with Design Automation Algorithms

1 State of the art and preliminary work

Microfluidic biochips are revolutionizing the traditional biochemical experiment flow with their high execution efficiency and miniaturized fluid manipulation [1, 2, 3]. Devices are built in such a chip to execute specific operations, such as mixing and detection. Fluid samples are transported through microchannels between devices to carry out the protocol of a bioassay. All these functions are performed at the nanoliter level and controlled by a microcontroller without human intervention. The efficiency and reliability of such miniaturized and automated chips endow them with a great potential to improve human life significantly, and the research to bridge them with real-world applications is key to their success.



(a)

(b)

(c)

Figure 1: Components and structure of flow-based biochips. (a) Valves constructed at intersections of flow/control channels [3]. (b) Mixer [3]. (c) A part of a biochip containing a mixer surrounded by a transportation channel network [4].

A flow-based microfluidic biochip is constructed from basic components such as microchannels and microvalves, henceforth named as channels and valves for simplicity. Flow channels are used to transport reaction samples and reagents between different locations. Above flow channels, control channels are built to conduct air pressure to intersections of flow channels and control channels to form valves, as illustrated in Figure 1(a), where three valves are constructed at the intersections. These channels are built from elastic materials, so that air pressure in a control channel can block the movement of fluid sample by squeezing the flow channel downwards. Conversely, if the pressure in the control channel is released, the fluid sample can resume its movement. Since the channel width has been miniaturized down to 50 um [5] thanks to the advance of manufacturing technology, a huge number of channels and valves can already be integrated into a single biochip to perform large-scale experiments and diagnoses.

With valves as basic controlling components, complex devices can be constructed. For example, mixers can be built using channels and valves to execute mixing operations, which are very common in biochemical applications. The structure of a mixer is shown in Figure 1(b), where the three valves at the bottom are actuated alternately by applying and releasing air pressure in the control channels to mix fluid samples and reagents by peristalsis. The execution of a mixing operation in a biochip is demonstrated in a video [6]. After the mixing operation is completed, the resulting fluid sample can be stored in a dedicated storage unit temporarily.

In a biochip, devices executing specific operations, e.g., mixing and heating, are connected by channels so that intermediate reaction results can be transported between devices for processing. All these operations and transportation are controlled by a microcontroller, which issues instructions in a given order to actuate valves to move fluid samples and execute operations. Figure 1(c) shows a mixer (reaction loop) surrounded by flow channels (green), control channels (yellow and red) and valves (yellow and red blocks). These channels and valves together form a network similar to the road transportation system. If fluid channels should cross, valves are built to form a switch, as shown in Figure 1(c). At any moment, only two out of the four valves should be opened to direct fluid transportation; the other two valves must be closed to avoid fluid contamination. Consequently, the role of the valves at the intersection of flow channels is similar to that of the traffic lights in the road transportation system, while the open/closed states of the valves are controlled by a microcontroller according to the protocol of the application. The mixer and the channel network in Figure 1(c) are implemented into a biochip of the size comparable to that of a coin as shown at the upper left corner, demonstrating the miniaturized integration of microfluidic biochips.

In a biochip, the open/closed states of valves and the transportation of fluid samples are determined according to the biochemical application executed by the biochip. A biochemical application, or *bioassay* henceforth, is usually described with a *sequencing graph* $\mathcal{G} = (\mathcal{O}, \mathcal{E})$, such as in Figure 2, where $\mathcal{O}$ is the set of nodes and $\mathcal{E}$ is the set of edges. A node $O_i \in \mathcal{O}$ in the sequencing graph represents an operation, whose type τ_i and duration u_i are specified by the user. The type τ_i of the operation, e.g., mix, heat and filter, is predefined by the application. To execute an operation, the corresponding device must be built in the chip and the operation must be assigned to this device. An edge $e_{ij} \in \mathcal{E}$ from O_i to O_j in the sequencing graph specifies that O_i must be executed before O_j and the result of O_i is the input of O_j. If O_i and O_j are executed by different devices, the required fluid transportation must be performed by the channel network between devices.

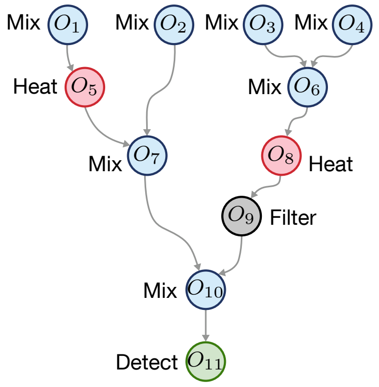


Figure 2: Sequencing graph of a bioassay.

Biochips have a huge advantage over the traditional manual experiment flow, where operations performed by humans are error-prone and inaccurate. Any inadvertent mistake in this manual process might ruin a complex experiment that may last for several days. In a biochip, the volumes of fluid samples and reagents are controlled accurately and fluid samples are moved to target devices reliably, all of which are managed by a microcontroller exactly following a given protocol. Furthermore, reactions with fluid samples and reagents of tiny volumes take less time to complete than those with large volumes in tubes and pipettes in the traditional experiment flow, so that biochips are also more responsive in dealing with urgent situations. Moreover, biochips save precious reagents by performing operations at nanoliter level. For example, RNase inhibitor, a polyclonal antibody commonly used in reverse transcription polymerase chain reaction, cost 600 euros per milliliter in December 2014 [7].

The miniaturization of microfluidic biochips also has the potential of large-scale system integration. Already in 2008, a biochip array with 25K valves was accomplished [8], and recent advances in manufacturing technologies have led to a valve density of 1 million per cm^2 [9]. A system integration of this scale enables long-aspired exhaustive diagnoses in identifying the illness of a patient by testing pathological samples with thousands of reagents simultaneously. This breakthrough will not only reduce the inaccuracy in medical diagnoses, where individual expertise and experience of doctors play an important role, but also change the current guess-then-test model of medical treatment. In addition, such exhaustive diagnoses can be performed in small health-care centers routinely, due to the tremendously miniaturized chip size and lowered cost. With this exhaustive diagnosis model, illnesses can be detected at a very early stage and treatment cost can be reduced significantly as well.

Owing to their efficiency and cost-effectiveness, microfluidic biochips are reshaping many fields such as pharmacy, biotechnology and health care. In recent years, genomic bioassay protocols, such as nucleic-acid isolation, DNA purification and DNA sequencing, have been successfully demonstrated with microfluidic biochips. In addition, this technology has attracted a lot of commercial attention, e.g., from Illumina [10], a market leader in DNA sequencing. Accordingly, the International Technology Roadmap for Semiconductors (ITRS) 2015 [11] has recognized the importance of microfluidic devices as having a rapid growth in the next several years.

Synthesis of microfluidic biochips using computer algorithms

In a biochip, operations are executed by a given number of devices with time multiplexing, described as a schedule. For example, an execution of the bioassay illustrated in Figure 2 is shown in Figure 3(a), where two mixers, one heater, one filter, and one detector are available. According to the schedule, the layout of a biochip, including the locations of devices and the transportation channels between them, can be determined to generate a physical design, as shown in Figure 3(b), where the devices are connected by a channel network controlled by valves.

The synthesis process above demonstrates that the schedule of operations of a bioassay determines the overall execution time. In addition, the fluid transportation between devices affects the structure of the channel network. Consequently, a holistic design automation flow is required to bridge the low-level components introduced by the microfluidics community with high-level real-world applications. In each step of this design automation flow, various design objectives should be optimized to achieve an efficient architecture for the biochip.

The synthesis flow of biochips is similar to the synthesis flow for integrated circuits [12]. Therefore, researchers in the electronic design automation community have started to expand into this area in recent years [13, 14]. However, these research efforts are still in an early stage and many unique characteristics of microfluidic biochips have still not been considered.

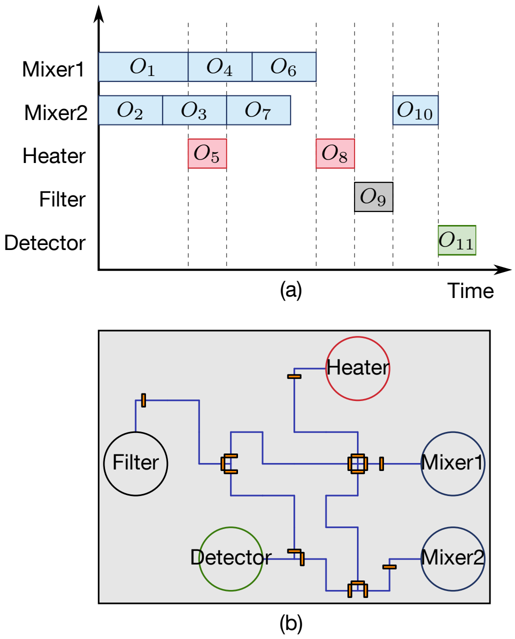


Figure 3: Synthesis of microfluidic biochips. (a) Scheduling. (b) Physical design.

Flow-based microfluidic architectures: the electronic view

In the microfluidics community, researchers are focusing on developing new technologies and new structures to build fundamental components and devices, such as valves and pumps [15, 16]. Prototype microfluidic biochips are also built very often to demonstrate the function and performance of new components and new devices. Another major focus of the microfluidics community is to increase the integration density of basic components. With the advance in MEMS technology, a large number of components such as valves can now be built in a single biochip [9]. Unfortunately, the abundant available resources have mostly been left unexplored, because end users cannot use them without a system layer that presents an interface for user applications, similar to the scenario that an operating system is missing for computer users. On the other hand, the effort of the microfluidics research community has been spread out in exploring even more technologies for microfluidic biochips, leading to a flourishing but fragmented panorama in the research on microfluidics.

The status of the microfluidics community is similar to the early period of the semiconductor industry. At that time, researchers were exploring different materials and device structures to build smaller but faster transistors. Thereafter, CMOS-based technology became dominant in this industry, while other technologies are employed only for specific applications. CMOS technology obtained its dominance because of its performance. However, the development of Electronic Design Automation (EDA) has supported the large-scale integration in design and manufacturing and made the computing resources available to designers successfully.

Observing the state of the art of microfluidic biochips, researchers from computer science and electrical engineering have started to bring their own computing models into microfluidic biochips. For example, the architecture of a microfluidic biochip from [17] is shown in Figure 4. In this architecture, the mixer functions as the computing unit and intermediate results from the mixer are stored in the dedicated storage unit. The cells in the storage unit are built from normal channels. At the ports of this storage unit, valves form multiplexers to direct fluid samples to enter into or leave from specific cells. This architecture emulates the classical von Neumann computer architecture to build a biochemical computing system from basic components. However, this simple emulation forsakes many unique characteristics of flow-based biochips, leading to inefficient execution of bioassays.

Similar to the semiconductor industry, design automation tool chains are also needed to support the development of microfluidic biochips. In recent years, the electronic design automation community has tried to migrate the existing design methodologies for integrated circuits to microfluidic biochip design, covering the phases from high-level synthesis down to physical design. Although this top-down flow has served the semiconductor industry in the past 50 years very successfully, fundamental changes should still be made to deal with specific requirements of biochips and take advantage of their unique features.

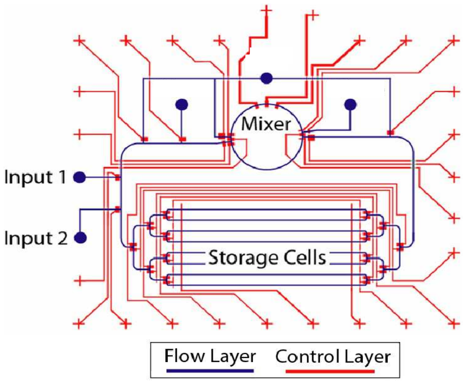


Figure 4: Computing-based biochip architecture containing a mixer and a dedicated storage unit with eight cells [17].

Flow-based microfluidic biochips: the unique characteristics

In microfluidic biochips, the inputs to an operation are fluid samples. Unlike electrical signals in integrated circuits, these fluid samples have a physical mass. In executing operations of a bioassay, fluid samples are processed with various operations, such as mixing, heating and detecting in different devices. The results of these operations are often fluid samples of different properties, so that inadvertent contamination between them should be avoided. The intermediate results of these operations should be stored in the chip temporarily in case they are not used immediately. Consequently, the physical mass and the variety of fluid samples become the major differences between biochips and integrated circuits, leading to several unique characteristics in biochip design.

Volume Management: In executing a bioassay, the volumes of fluid samples should be managed. Assume all the devices executing the bioassay in Figure 2 have a capacity ν. Each of the resulting samples of O_1 and O_2 thus has a volume ν. When these two fluid samples reach the device executing O_7, half of their volumes should be disposed of because the device only accepts a volume ν. This volume management is not stated explicitly in the sequencing graph, but must be dealt with implicitly according to the volumes of intermediate fluid samples and the capacities of devices.

Storage management: In the schedule in Figure 3(a), O_2 completes before O_5 does. The intermediate result of O_2 should be moved out of Mixer2 and stored somewhere temporarily so that the mixer can execute O_3. In the biochip shown in Figure 4, this storage function is fulfilled by moving the result of O_2 to the dedicated storage unit through a channel. In synthesizing biochips, if operations are not scheduled properly, many storage requirements may appear, leading to many transportation channels and a large storage unit. In contrast to a dedicated storage unit as shown in Figure 4, the storage function can actually be implemented using distributed transportation channels. In fact, a fluid sample can stay anywhere in a channel in the biochip until it is used by the next operation. This is a significant difference between biochips and electronic systems, where intermediate data can only be stored in special memory units, either flip-flops or RAM components. This observation can be confirmed by the storage cells in the dedicated storage unit in Figure 4. These cells are built of normal channels but with valves at each end of a channel to control the store/fetch operations. Instead of forming a monolithic storage unit, these channels and valves can actually be distributed in the chip so that they can be used for storage when required, and for transportation otherwise, leading to better flexibility and wearing balancing.

Washing: Unlike electrical signals, fluid samples leave residue in channels after they travel through them. Before such a channel is reused by another fluid sample, it should be washed by neutral fluids such as silicon oil. Washing contaminated channels can be very flexible because several channel segments can be washed simultaneously if they form a connected graph while being isolated from the rest of the biochip that is executing other operations.

Flow-layer and control-layer codesign: In a flow-based biochip, valves are controlled by air pressure through control channels, e.g., the red channels in Figure 4. If all the valves are controlled independently, the routing of control channels in a complex design becomes very complicated. To solve this problem, control channels of some valves can be shared if operations can still be executed correctly. This sharing requires a codesign of the flow layer and the control layer to match the actuation patterns of valves.

State-of-the-art research on design automation for flow-based biochips using computer algorithms

Several methods have been proposed to synthesize flow-based biochips. The method in [18] proposes a top-down flow to generate a biochip architecture while minimizing the execution time of a bioassay. The flow channel routing problem considering obstacles is solved with an algorithm based on rectilinear Steiner minimum tree in [19]. These methods still assume that intermediate fluid results can be stored automatically in a dedicated storage unit as in the biochip inspired by electronic design shown in Figure 4. The real storage process and its efficiency, however, have not been investigated.

To avoid contamination, a method based on path searching is introduced in [20] to wash devices and channel segments. This method still traces path sets and block-based partial washing has not been explored. The latter requires a co-optimization between operation scheduling and washing activities.

Control logic synthesis is investigated in [21] to reduce the number of control pins. The method in [22] minimizes pressure-propagation delay in the control layer to reduce the response time of valves and synchronize their actuations. Furthermore, flow layer and control layer codesign is investigated in [23] to achieve valid routing results on both layers iteratively, and length-matching in routing control channels is considered in [24] as well. Since these methods do not consider operation scheduling, the number of control channels may still be large and consequently they might not be routed successfully.

Though the volume management problem in biochips has been explored as early as in 2008 [25], and later in [26] for the specific bioassay sample preparation, the optimization of volume management for general bioassays and the interaction of this task with fluid transportation for normal operations have not been taken into account.

When the unique characteristics of biochips are considered, the tasks of synthesizing flow-based biochips are entangled with each other. Consequently, a systematic design flow covering architectural synthesis, control layer design, washing and volume management should be explored, which is the major objective of this project.

Preliminary work

Observing the great potential of microfluidic biochips and the design automation challenges at the eve of their large-scale integration, I have initiated the research on biochips in the Institute for Electronic Design Automation at TUM. Applying the knowledge on design automation methods for IC design to microfluidic biochips, several preliminary ideas have been verified in our research group to synthesize efficient biochip architectures.

In the research community, we have pioneered the idea of distributed channel storage in flow-based biochips [27]. We have also proposed to improve the reliability of biochips with a large-scale integration [28], where a fully reconfigurable valve array is used to execute operations and fulfill the functions of transportation and storage. Furthermore, we have introduced a path-based vector generation method for test of microfluidic biochips [29]. Fault localization and design-for-testibility of microfluidic biochips have been addressed in [30, 31]. Moreover, optimization of control logic to improve its efficiency and the overall portability of the microfluidic platform has been explored in [32].

To bridge microfluidic biochips with their applications, techniques are required to map bioassays to specific architectures. More importantly, the structures of bioassays may influence biochip architectures because different sequencing graphs lead to different execution, transportation and storage requirements. Therefore, efficient algorithms are also needed to optimize biochip architecture and assay execution. In the past, our research group and the EDA institute had broad research activities in design automation for integrated circuits with well-recognized results. The developed algorithms and tools may also potentially benefit the research on microfluidic biochips, e.g., those for physical design [33], circuit test and tuning [34] (Nomination for Best Paper Award at DAC 2016), reliability [35], as well as hierarchical modeling and analysis [36].

1.1 Project-related publications

1.1.1 Articles published by outlets with scientific quality assurance, book publications, and works accepted for publication but not yet published.

1. T.-M. Tseng, **B. Li**, U. Schlichtmann, and T.-Y. Ho, ‘‘Storage and caching: Synthesis of flow-based microfluidic biochips,’’ *IEEE Design & Test*, vol. 32, no. 6, pp. 69–75, 2015.
2. T.-M. Tseng, **B. Li**, T.-Y. Ho, and U. Schlichtmann, ‘‘Reliability-aware synthesis for flow-based microfluidic biochips by dynamic-device mapping,’’ in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2015, pp. 141:1–141:6.
3. T.-M. Tseng, **B. Li**, M. Li, T.-Y. Ho, and U. Schlichtmann, ‘‘Reliability-aware synthesis with dynamic device mapping and fluid routing for flow-based microfluidic biochips,’’ *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 35, no. 12, pp. 1981–1994, 2016.
4. T.-M. Tseng, M. Li, **B. Li**, T.-Y. Ho, and U. Schlichtmann, ‘‘Columba: Co-layout synthesis for continuous-flow microfluidic biochips,’’ in *Proc. ACM/IEEE Design Automation Conference (DAC)*, 2016, pp. 147:1–147:6.
5. Q. Wang, Z. Li, H. Cheong, O. Kwon, H. Yao, T.-Y. Ho, K. Shin, **B. Li**, U. Schlichtmann, and Y. Cai, ‘‘Control-fluidic codesign for paper-based digital microfluidic biochips,’’ in *Proc. IEEE Int. Conf. Comput.-Aided Des. (ICCAD)*, 2016, pp. 103:1–103:8.
6. Q. Wang, S. Zuo, H. Yao, T.-Y. Ho, **B. Li**, Ulf Schlichtmann, and Y. Cai, ‘‘Hamming-distance-based valve-switching optimization for control-layer multiplexing in flow-based microfluidic biochips,’’ in *Proc. IEEE Asia and South Pacific Des. Autom. Conf. (ASP-DAC)*, 2017, pp. 524–529.
7. C. Liu, **B. Li**, B. B. Bhattacharya, K. Chakrabarty, T.-Y. Ho, and U. Schlichtmann, ‘‘Testing microfluidic fully programmable valve arrays (FPVAs),’’ in *Proc. IEEE Design, Autom., and Test Europe Conf. (DATE)*, 2017, pp. 91–96.
8. C. Liu, **B. Li**, T.-Y. Ho, K. Chakrabarty, and U. Schlichtmann, ‘‘Design-for-testability for continuous-flow microfluidic biochips,’’ in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2018, pp. 164:1–164:6.
9. Y. Zhu, **B. Li**, T.-Y. Ho, Q. Wang, H. Yao, R. Wille, and U. Schlichtmann, ‘‘Multi-channel and fault-tolerant control multiplexing for flow-based microfluidic biochips,’’ in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, 2018, pp. 123:1–123:8.
10. A. Bernardini, C. Liu, **B. Li**, and U. Schlichtmann, ‘‘Fault localization in programmable microfluidic devices,’’ in *Proc. IEEE Design, Autom., and Test Europe Conf. (DATE)*, 2019.

1.1.2 Other publications

None

1.1.3 Patents

1.1.3.1 Pending

None

1.1.3.2 Issued

None

2 Objectives and work programme

2.1 Anticipated total duration of the project

Three years

2.2 Objectives

The major objective of this project is to develop computer algorithms for design automation of microfluidic biochips. Based on fundamental components such as valves and devices and design constraints, efficient microfluidic architectures will be synthesized automatically. The resulting design framework will be capable of offloading design tasks from chip designers by computer algorithms and assisting them to explore new chip designs efficiently.

From the perspective of computer-aided design, synthesis of microfluidic biochips can be partitioned into three levels. At the lowest level, valve activities and flow transportation are investigated. The task of this step is to generate an efficient chip architecture for a given bioassay or multiple bioassays automatically. The second level of synthesis deals with special requirements of washing and flow-/control-layer codesign. The result of this synthesis is overlaid upon the biochip architecture generated in the previous step to form a complete architecture-schedule solution. Since the first two steps focus on concrete behavior of basic components such as valves and channels, they are called *microsynthesis* henceforth. Biochips considered in microsynthesis are relatively small-scale due to the scalability challenges in optimizing exact switching activities of all valves and flow transportation tasks. These biochips are very useful in small biochemical labs or in the scenarios such as point-of-care test.

With the advance of manufacturing technology, the integration density of biochips has increased tremendously. Correspondingly, a bioassay executed by a biochip of this scale also contains a very large number of operations for resource-intensive applications, e.g., exhaustive diagnoses for illness detection. Therefore, synthesis of biochips at the highest level, or macrolevel, thus called *macrosynthesis* henceforth, deals with the challenges considering a large-scale system integration.

The synthesis techniques of the three levels form a comprehensive tool chain for flow-based biochips. The relation of these tasks from architectural synthesis to cyber-physical integration is illustrated in Figure 5. Since it is challenging for designers to manually deal with these tasks for complex microfluidic systems, the goal of this project is thus to develop design automation algorithms to assist designers to deal with them. This scenario is similar to the case that EDA research contributes in the general scope of the semiconductor industry, where it assists designers with software tools to manage the exploding design details of complex ICs.

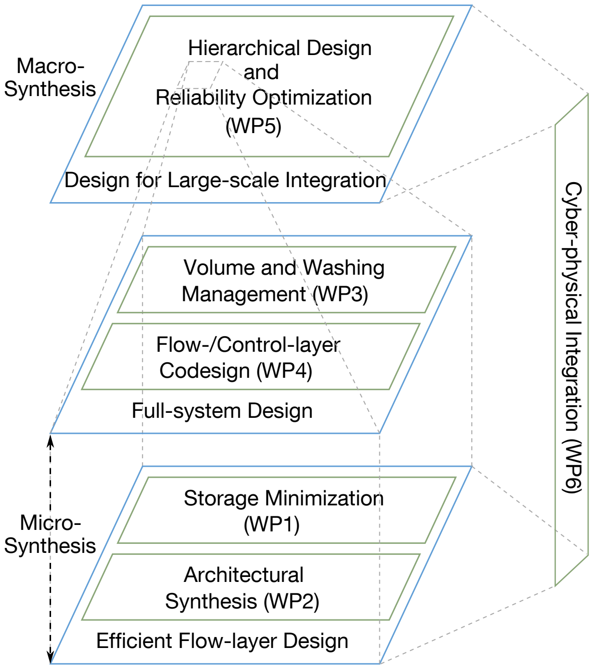


Figure 5: Synthesis levels for biochips.

In this project, we will focus on micro- and macrosynthesis of biochips to generate efficient biochip designs automatically. The unique characteristics of biochips discussed in the previous section will be incorporated into this synthesis process. Furthermore, the interaction of the automatically generated biochip designs with sensors and cameras in cyber-physical systems will be investigated. This project will demonstrate the effectiveness of design automation algorithms in improving the performance and reliability of biochips. More importantly, it will bridge microfluidic technologies with real-world applications using the developed tool chain to realize the great potential of microfluidic biochips.

2.2.1 Microsynthesis of microfluidic biochips

Since microsynthesis deals with compact biochips that only contain a small number of devices, operations need to be arranged along the timeline to reuse devices. Consequently, operations are assigned in queues at devices, such as the schedule in Figure 3(a). With this schedule, a biochip architecture is generated automatically that carries out the operations and the transportation tasks between devices. To reduce the execution time of the bioassay, operations should not wait at devices unnecessarily longer than required. In addition, transportation tasks should affect the execution of operations as little as possible by moving fluid samples to target devices on time.

Storage minimization during scheduling

When devices in a biochip execute operations according to a schedule, the resulting fluid samples of some operations may not be used immediately. Consequently, these fluid samples need to be stored in the chip temporarily. The durations of these storage requirements affect the overall execution time of the bioassay, because improper storage may block the transportation of other fluid samples. In addition, a large number of simultaneously stored fluid samples consume many resources and require a large area due to their physical volume.

Assume that the pure transportation time from a device to another device in a biochip is a constant u_c. If an intermediate fluid sample needs to wait for a time larger than u_c before it is used, it must be stored somewhere in the chip temporarily. Figure 6(b)–(c) illustrates two schedules of executing the bioassay in Figure 6(a) using two mixers. In Figure 6(b), operation O_2 is scheduled before O_3. Since the result of O_2 must be stored until it is used by O_4 and O_5, half of the mixing result of O_2 needs to be transported to a location for storage $O_2 \rightarrow O_4$, which starts u_c time after O_2 finishes. Thereafter, the other half of the mixing result is moved out to initiate storage $O_2 \rightarrow O_5$. Consequently, two storage requirements are generated from this schedule. Since the lifespans of these storage requirements overlap, the storage capacity of the biochip should be at least two units. To reduce the storage requirement, O_3 can actually be scheduled before O_2 as shown in Figure 6(c). In this schedule, half of the result of O_2 is transported to Mixer1 and O_5 starts after the storage $O_3 \rightarrow O_5$ is fetched into Mixer2. This improved schedule generates only a single storage requirement, while the overall execution time of the bioassay is not changed.

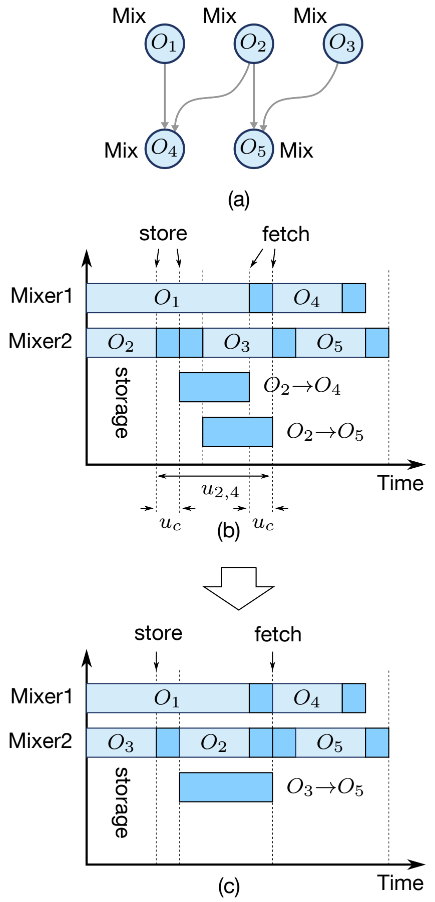


Figure 6: Storage reduction. (a) Sequencing graph. (b) Schedule with two storage requirements. (c) Schedule with one storage requirement. The execution times of the bioassay with these two schedules are equal.

In this example, we can observe that the lifespan of a stored fluid sample is determined by the difference between the ending time of the parent operation O_i and the starting time of the child operation O_j, denoted by $u_{i,j}$, e.g., $u_{2,4}$ in Figure 6(b). To reduce storage requirements, we introduce a *storage minimization objective* and minimize the intermediate storage requirements when scheduling operations to devices by solving

$$\text{Minimize} \quad \alpha t^E + \beta \sum_{e_{ij} \in \mathcal{E} \bigwedge d_i \neq d_j} u_{i,j} \tag{1}$$

where t^E is the total execution time of the bioassay similar to the definition in [12], $\mathcal{E}$ is the set of the edges in the sequencing graph, and $e_{ij} \in \mathcal{E}$ limits storage minimization to operation pairs with a direct edge in the sequencing graph. d_i and d_j represent to which devices operations O_i and O_j are assigned, respectively. Therefore, $d_i \neq d_j$ excludes the operation pairs assigned to the same device where no transportation is needed. The two constants α and β control the priority of the optimization objectives. The other constraints of this optimization problem are the same as those defined for high-level synthesis of digital circuits [12]. After solving (1), a schedule like Figure 6(c) instead of 6(b) is generated as the input to architectural synthesis in the next step.

During scheduling, if no device reuse is allowed in the biochip, a device can only execute one operation. The technique above, however, can still be applied to determine the relative execution relation of operations more efficiently than manual design.

Architectural synthesis with distributed channel storage

The schedule of a biochip specifies when and by which devices operations are executed. In addition, it states when and where transportation requirements between devices appear. To carry out operations and fulfill transportation tasks efficiently, the relative locations of devices as well as the channels connecting them should be determined. The types of the devices, e.g., mixers, heaters and filters, correspond to the types of the operations to be executed. These types are specified by the designer of the bioassay but only the dimensions of the devices influence their placement and the following routing of transportation channels.

When fluid samples are moved along channels, they can stop the movement and stay in a channel segment temporarily to avoid a transportation conflict, as cars stop before a red traffic light to let cars of the other direction pass. Namely, fluid samples can be cached in any channel in the biochip, provided that the channel is not used for other transportation tasks. After the conflict is resolved, fluid samples can resume the movement toward target devices. This *distributed channel storage* mechanism, or *caching*, is a unique characteristic of biochips compared with electronic systems, where data must be stored in memory units such as RAM instead of on interconnect wires.

This distributed caching mechanism allows a high efficiency of storing and fetching of fluid samples by avoiding access congestion at the ports of a dedicated storage unit. In addition, accessing on-the-spot distributed channel storage is more convenient because fluid samples can be cached close to the devices. Furthermore, this feature enables a chip architecture with a unified representation of transportation and storage channels to improve channel efficiency.

Since the flow layer in a biochip is only two-dimensional, intersections between flow channels in a complex system usually cannot be avoided. At each intersection, a switch needs to be built to direct the transportation flow to the target device. A switch is constructed by four valves at an intersection, as shown in Figure 7(a). At any given moment, two out of the four valves in a switch are opened to connect two channel segments. Consequently, a fluid sample needs to travel through several channel segments connected by switches to reach the target device. Correspondingly, these channel segments form a *transportation path* between two devices.

Considering devices and channels together, the architecture of a biochip can be described as devices connected by a network of channel segments. For example, the architecture of a biochip with five devices is shown in 7(b), where solid nodes represent switches and circles represent devices. Transportation paths between devices are formed from channel segments and connected by switches. Since transportation paths are used only when there is a fluid sample traveling through them, channel segments can be reused by different paths to increase efficiency. For example, path 1 and path 2 in Figure 7(b) share a common channel segment which they use at different times.

On the network of channel segments, distributed channel storage can be formed to replace the dedicated storage unit in Figure 4, which suffers access congestion at its ports. For example, in Figure 7(c), along path 3 a fluid sample is moved to the channel segment between A and B. However, the next operation using this fluid sample is scheduled later, so that the fluid sample must stay in the channel segment. During the lifespan of this storage, the channel segment between A and B cannot be used by other paths and the valves at the two ends of this channel segment must be closed. Consequently, other transportation tasks between devices need to be fulfilled by paths that do not include this channel segment, such as path 4 and 5 in Figure 7(d). When the cached fluid sample is finally needed, it is moved to the target device again by a newly constructed transportation path, which is shown as path 6 in Figure 7(e).

The concept of distributed channel storage unifies transportation and storage. Therefore, we can synthesize the architecture of a biochip from its schedule using a virtual *connection grid* as shown in Figure 7(f). At each node on the grid, either a device or a switch can be assigned. An edge on the grid represents a channel segment capable of caching a fluid sample. The task of architectural synthesis thus becomes to determine locations of devices and build time-multiplexing transportation channels on the grid. After architectural synthesis, all unused channel segments and switches in the virtual connection grid will be removed to reduce chip area.

In architectural synthesis, we need to reduce resource usage while guaranteeing that the schedule can be executed correctly. Therefore, we describe the architectural synthesis task as an optimization problem, as

Minimize Number_of_channel_segments (2)

s.t. Each device is assigned once on the grid; (3)

Simultaneous transportation paths should not intersect; (4)

Channel storage should be isolated from other paths; (5)

Scheduled operations are executed on time. (6)

The non-intersecting constraint (4) for a simple path, e.g., p_r in Figure 7, can be expressed as that the number of edges incident to a node on the path should not exceed two. However, it becomes more complex when a storage is involved. For example, the sub-paths $p_{r,1}$, $p_{r,2}$ and $p_{r,3}$ in Figure 7(f) represent the three phases of a channel storage. While $p_{r,1}$ and $p_{r,3}$ can be considered as normal transportation paths, $p_{r,2}$ should be processed specially because the two ending nodes of the channel segment can be used by paths conducting other transportation tasks, such as p'_r in Figure 7(f).

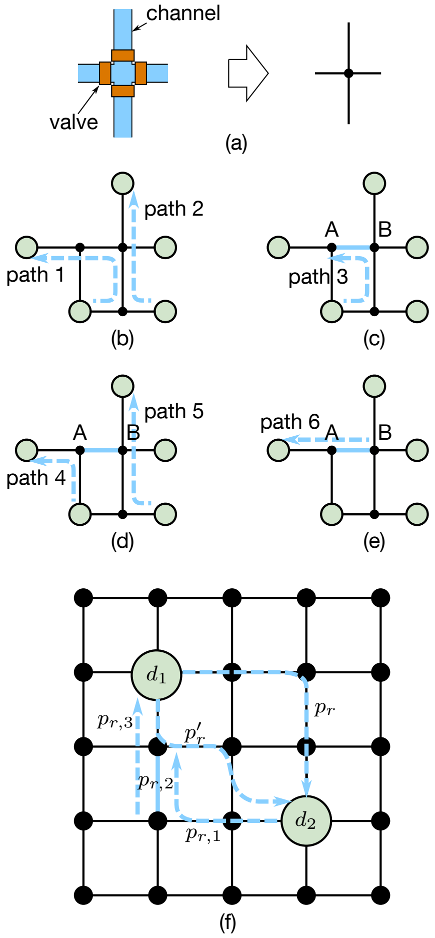


Figure 7: Switch and channel storage. Circles represent devices and solid nodes represent switches. (a) Switch structure. (b) Two transportation paths sharing one channel segment with time multiplexing. (c) Fluid sample to channel storage. (d) Storage in channel segment. (e) Fluid sample from channel storage to device. (f) Connection grid for architectural synthesis.

In solving the optimization problem, the minimization of the number of channel segments built in the biochip also generates a compact architecture, because devices and edges used to construct transportation paths tend to be close to each other due to the optimization objective, no matter whether devices can be reused in executing operations or not. Afterwards, the locations of devices are determined as indicated by the variables associated to the nodes.

Since determining an optimal chip architecture on a connection grid is an NP-hard problem due to the complex combinations of device locations and their connections, heuristic strategies need to be deployed. For example, the relative locations between devices can be preconfigured according to the transportation requirements between samples produced by them. In addition, nonregular connection grid structures may be explored to reduce the computational effort.

Volume management and washing

A special characteristic of biochips is that volumes of fluid samples should be managed during execution of a bioassay. For example, in Figure 2, the mixer executing O_{10} can accept only half of the volumes from O_7 and O_9, and the other half needs to be disposed of to the waste. This volume management should be dealt with very often in diagnosis-purposed bioassays, where usually only the result of the final detection operation is important. In practice, there is still flexibility in managing fluid volumes. For example, a half of the result from O_8 can be disposed of before O_9 or after O_9. In synthesizing a biochip, these implicit volume management requirements as well as the optimal time to deal with them should be identified by the synthesis tool automatically. Thereafter, additional flow paths should be constructed to move superfluous volumes to the waste port, such as the waste path shown in the upper left corner in Figure 8.

Another implicit requirement in biochip synthesis is washing. During the execution of a bioassay, after some channel segments are used either by normal transportation paths or waste paths, they should be washed before they can be used for normal fluid transportation again. However, washing operations can be postponed if there are still enough clean channel segments to construct new transportation paths. Consequently, channel segments can be combined into as few washing tasks as possible to improve washing efficiency.

Unlike flow paths that transport fluid samples and reagents, washing can be performed blockwise instead of pathwise. For example, at a given moment, the valves on some switches can be closed simultaneously to form a wall, as shown by the red dashed line in Figure 8. Afterwards, the left half of the biochip can be flushed at the same time. Combined with postponed washing operations, this technique can increase washing efficiency significantly by collecting used paths together and washing them as a whole, instead of performing traditional pathwise washing operations. Furthermore, fluid disposal can be combined with washing operations, because superfluous fluid to be disposed of and washing fluid all go to the waste finally. Consequently, a global strategy should be developed to schedule the waste paths together with the introduced block-based washing, or flushing, to reduce interference between these auxiliary functions and the normal bioassay operations.

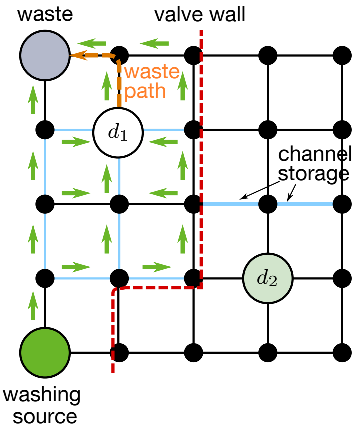


Figure 8: Volume management and washing.

Flow-layer and control-layer codesign

Valves in a biochip are controlled by air pressure through control channels. By switching between a high air pressure and a low air pressure in a control channel, a valve can be closed or opened. In a large design, there may be many valves. If for each valve an independent control channel is constructed, the routing complexity becomes very demanding. In addition, for each independent control channel, an air pressure source has to be provided, leading to an increase of cost.

To reduce the number of independent valves, the patterns of valve switching can be synchronized. If two valves always share the same switching pattern, they can be connected to the same control channel. In real designs, this synchronization is very often not possible, because even with a single mismatch in the switching patterns of two valves, their control channels cannot be merged. To alleviate this problem, the switching patterns can be aligned by examining each mismatching switching of two valves. If the state of a valve can be set to the state of the other valve without affecting the execution of the bioassay, this mismatching can then be resolved.

Since valve switching patterns are determined by the operations and transportation tasks, a codesign between flow layer and control layer should be performed. For example, there are three valves inside a mixer to mix the fluid samples by peristalsis, as shown in Figure 1(b). These valves are also shown in Figure 9(a), which is a part of the biochip in Figure 3(b). If the three valves in each of these two mixers always switch with the same pattern, they can share the same set of control channels. In reality, this is a rare case since the operations executed by these mixers are usually not aligned in the schedule as shown in Figure 6(c), where the valves in Mixer1 continue to switch after Mixer2 finishes the execution of the operation O_3.

To improve the alignment of the valve switching activities, operations can either be paused or their durations can be extended provided that this extension does not affect the results. For example, the schedule in Figure 6(c) can be revised to create a new schedule shown in Figure 9(b). In this case, O_1 is paused when the result of O_3 is moved out of Mixer2. When O_2 starts, O_1 resumes its execution. The execution of O_2 is also extended so that its ending time is aligned with the ending time of O_1. Afterwards, O_4 is delayed to align its starting time with the starting time of O_5. With these changes, the switching patterns of valves in the two mixers are completely aligned so that they can share the same set of control channels.

In flow- and control-layer codesign, the overall execution time of the bioassay may be prolonged as illustrated in Figure 9(b). Consequently, in revising the schedule of a bioassay, an objective of optimization is to reduce the extension of the overall execution time as much as possible, by carefully scheduling the operations to devices and arranging their execution sequences.

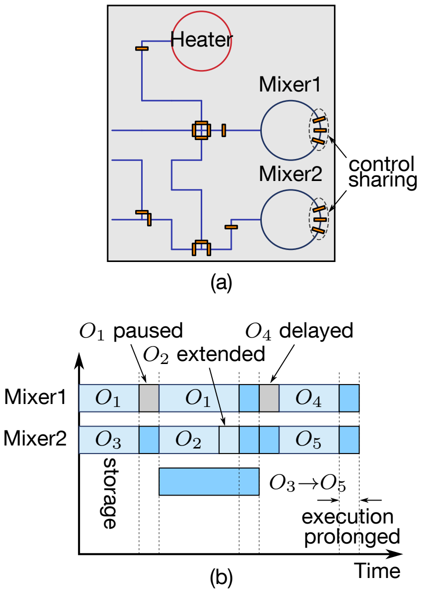


Figure 9: Flow- and control-layer cosynthesis.

2.2.2 Macrosynthesis of microfluidic biochips

Macrosynthesis processes a bioassay with a large number of operations for resource-intensive applications. The biochip executing such a bioassay also contains plentiful devices such as mixers. Due to the scalability challenges, it is very difficult to perform microsynthesis on an entire bioassay of this scale and complexity. Instead, the bioassay is partitioned into functional subgroups according to the relation between operations. For example, operations between which much fluid transportation exists should be grouped together. Thereafter, the function subgroups are considered as basic modules and mapped to the biochip as a floorplan. After this mapping, microsynthesis as introduced above is performed inside each subgroup to improve the execution efficiency of the operations. The basic concept of this hierarchical partitioning and synthesis is illustrated in Figure 10.

The objectives of synthesis at this level include: 1) operations in the sequencing graph should be partitioned in a way to reduce the communication between function subgroups; 2) the size of each subgroup should be balanced to reduce the complexity of microsynthesis inside each subgroup; 3) the floorplan of the biochip should determine locations of the subgroups, and subgroups requiring fluid transportation should be located in close proximity; 4) the floorplan should consider special devices such as sensors and heaters.

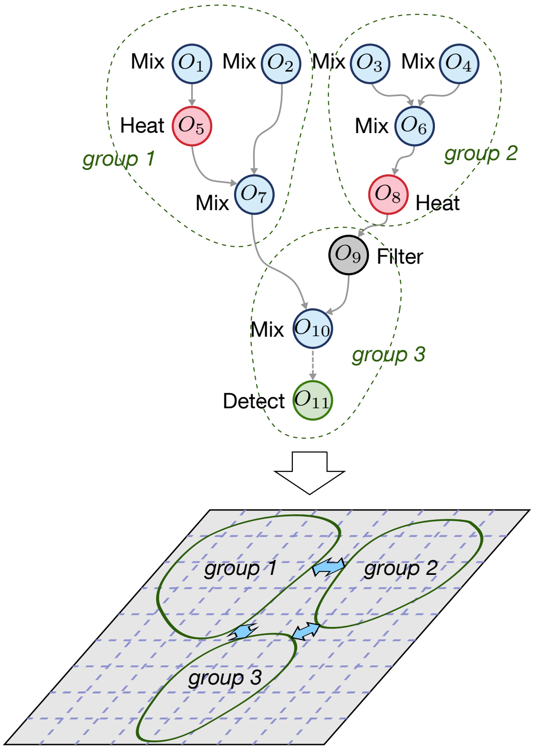


Figure 10: Synthesis at the macrolevel.

During macrosynthesis, the reliability of the biochip should also be considered. As demonstrated in the video [6], the three valves in the mixer switch much more often to drive the circular flow inside the mixer than the other valves that only control fluid transportation. Consequently, these valves wear out more quickly than the others. To maintain a reliable function of the biochip, operations should be distributed evenly among devices of the same type to lower the worst-case wearout. This reliability issue should be taken into account as early as during hierarchical partitioning to balance the mixing operations in the subgroups so that the overall reliability of the biochip can be improved.

2.2.3 Cyber-physical integration of microfluidic biochips

In the final phase of this project, the algorithms developed in the previous phases will be evaluated in the context of a test platform as illustrated in Figure 11. On this platform, sensors provide information about real-time status of fluid samples, e.g., locations and reaction stages, so that even operations with nondeterministic duration can be executed efficiently, such as that in [37] from the collaborators of this project. The movement of fluid samples can also be monitored by a camera and coarse-grained pattern recognition can be performed by the microcontroller to capture snapshots of the on-going execution. In addition, this cyber-physical integration provides information about the health of the system, such as leakage in channels due to manufacturing defects and aging. Once known, these defects can be circumvented by adjusting the schedule of the operations and the mapping of function subgroups, so that the chip can still work properly. In this phase, representative assays such as PCR (Polymerase Chain Reaction), IVD (In-Vitro Diagnostics) and CPA (Colorimetric Protein Assay) will be used to verify the integration of the overall framework from microsynthesis to macrosynthesis developed in this project.

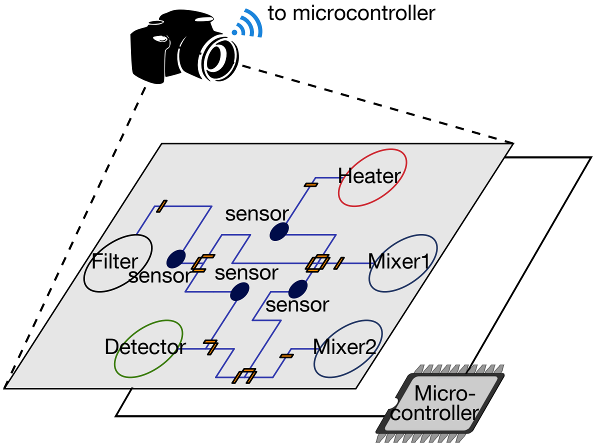


Figure 11: Integration of a synthesized biochip in a cyber-physical system.

2.3 Work programme incl. proposed research methods

The work plan of this project is divided into six packages (WP1–WP6), where WP1–WP4 belong to the microsynthesis phase, WP5 belongs to the macrosynthesis phase, and WP6 belongs to the cyber-physical integration phase. The work plan is shown in Figure 12, where "Stud." means student research assistant and "BA/MA" means bachelor/master thesis.

[Project Synthesizing Flow-based Microfluidic Biochips]						
Month:	1	7	15	20	25	31 … 36
Phases	Microsynthesis of biochips				Macrosynthesis	Cyber-phys. Int.
WPs	WP1	WP2	WP3	WP4	WP5	WP6
Stud.	Implementation and Test Assistance					
BA/MA		Microsynthesis	Control/Flow Codesign		System Integration	
Publications		P1		P2	P3	P4

Figure 12: Work plan and work packages.

The relation of these work packages is also shown in Figure 5 already. At the lowest level, storage minimization and architectural synthesis generate a complete biochip architecture with distributed channel storage. Afterwards, the results of volume management, washing, and flow-/control-layer codesign at the second level will be overlaid on the generated chip architecture to form a systematic solution for microsynthesis of biochips. The tasks of hierarchical design and reliability optimization deal with large-scale biochip systems, where the tools from microsynthesis are used to synthesize submodules in macrosynthesis. Finally, the integration of all the methods is tested in the cyber-physical integration phase.

2.3.1 WP1: Storage minimization during scheduling

In the first work package, storage requirements should be reduced during scheduling. Since the lifespan of a storage is determined by the ending time of a parent operation and the starting time of a child operation, as illustrated in Figure 6, storage minimization can be achieved by solving the optimization problem (1). The challenge of this task is that the minimization of the overall storage requirements does not guarantee that simultaneously existing storage requirements are the minimum, which in fact determines the required storage capacity that should be maintained in the chip. In this work package, the optimization problem (1) will be extended to take this objective into account. In addition, heuristic techniques such as extending the list-based scheduling algorithm in [12] will be explored to improve the execution efficiency.

This work package also includes literature exploration and the development of a basic framework for the project.

2.3.2 WP2: Architectural synthesis with distributed channel storage

With the result of scheduling, a biochip architecture can be generated automatically. In the second work package, a new architecture with distributed channel storage will be explored. This architecture unifies storage and transportation into a network of channel segments. Since each channel segment can switch its role between temporary storage and transportation, the efficiency of both storage and transportation can be improved.

This architecture is completely different from the classical von Neumann architecture as in Figure 4, so that a new model describing the new system behavior needs to be developed in this work package. To create this model, the constraints for, 1) device allocation on the connection grid shown in Figure 7, 2) construction of flow transportation paths, and 3) construction of temporary distributed channel storage without contamination, should be established.

With the constraints above, a framework for automatic architectural synthesis will be developed. In the first step, we formulate this synthesis task into an Integer Linear Programming (ILP) problem and find an optimal chip architecture using an ILP solver. This formulation will be improved with heuristic techniques such as preconfigured device locations and nonregular connection grid structures as well as hierarchical modeling to increase the scalability in the second step of this work package.

At the end of this work package, the first milestone of this project will be achieved, resulting in an efficient biochip architecture with distributed channel storage. We expect to publish the first paper from this project after finishing this work package.

2.3.3 WP3: Volume management and washing

In the third work package, the challenges of volume management and washing will be investigated. These challenges are unique for biochips as described in Section 2.2.1. Both challenges can be formulated as to construct additional flow paths at given moments. In this work package, we will develop a unified model to deal with these challenges simultaneously. As illustrated in Figure 8, the efficiency of volume management and washing can be improved by flushing an area of the chip instead of individual paths. We will investigate how the valve wall can be constructed and how the subtasks of volume management and washing can be grouped together for flushing in this work package.

2.3.4 WP4: Flow-layer and control-layer codesign

When the control layer in a biochip is considered, switching activities of valves in devices and in the transportation network should be aligned as much as possible to reduce the number of independent control channels. This alignment needs to be considered during both scheduling and architectural synthesis, as well as in volume management and washing. In the scheduling, operations can actually be arranged to align their starting times and they can also be paused or extended so that the switching activities of valves can be aligned. During architectural synthesis, flow samples are transported and stored in channel segments, so that they should also be considered to reduce mismatch in valve activities inside the transportation network. Furthermore, both volume management and washing need to open and close valves. These valve activities, however, have much flexibility because volume management and washing can be postponed if the ongoing operations are not affected.

The flow-layer and control-layer codesign can be considered as a fine-tuning step of the methods developed in WP1–WP3. The challenge is that the overall execution time of the bioassay should still meet the given specification after this fine-tuning.

At the end of this work package, a complete work flow for microsynthesis of flow-based biochips will be implemented. The result will be summarized and published as the second paper from this project.

2.3.5 WP5: Hierarchical synthesis of large-scale biochemical assays

While the increasing integration of biochips provides the computing power for large-scale biochemical analysis, such as future exhaustive tests and diagnoses in medical care, macrosynthesis of bioassays with a very large number of operations needs to be investigated closely. Due to their large number, operations in these bioassays cannot be considered individually during the synthesis process.

In the fifth work package, we will explore methods to partition large bioassays into function subgroups to reduce the synthesis complexity. This partitioning and mapping should minimize the transportation requirements between function subgroups. In mapping these function subgroups to a biochip, the floorplan should guarantee that the subgroups with much transportation are allocated not far away from each other to avoid unnecessarily long transportation channels.

For each subgroup, the microsynthesis flow developed in the first phase of this project will be used to optimize the internal structure to improve execution efficiency. We plan to publish the result of this work package as the third paper from this project.

2.3.6 WP6: Cyber-physical integration of microfluidic biochips

In practice, a biochip system may contain sensors and cameras to monitor the movement of fluid samples. In addition, applications may include operations with flexible durations. For example, some operations need to be stopped when the reaction reaches a given threshold. These information can be collected by sensors and cameras and sent to the microcontroller to adapt the valve switching patterns.

In the last work package, the algorithms developed in the previous work packages will be integrated and tested together with the collaborators. The data from sensors in the platform of the collaborators [37] will be analyzed and used to improve the synthesis flow introduced in this project. In addition, the chance to expand this work flow to incorporate further features and deal with challenges emerging from this system integration will be explored.

The last work package integrates the methods developed in WP1–WP5 to form a systematic solution for synthesis of flow-based biochips. The results of this project will be summarized and published as a journal article.

2.4 Data handling

The results of this project will be published at leading conferences and journals in the field of design automation for integrated systems.

To facilitate the access to these papers, they will also be published with the IEEE open access publication option, so that researchers around the world have free access to them. This option will enhance the reuse of the methods developed in this project and improve the interdisciplinary research on design automation for emerging systems.

2.5 Other information

None

2.6 Explanations on the proposed investigations

This project does not involve humans and human materials as well as animals in the experiments.

2.7 Information on scientific and financial involvement of international cooperation partners

In this project, we will collaborate with the research group of Prof. Krishnendu Chakrabarty from Duke University, US, and the research group of Prof. Tsung-Yi Ho from National Tsing Hua University, Taiwan.

Prof. Chakrabarty is one of the pioneers to apply computer algorithms to microfluidic design. His research ranges from application mapping to manufacturing testing, which has resulted in more than 550 papers published at top conferences and journals, in which more than 250 papers are on design automation for microfluidics (`http://people.ee.duke.edu/~krish/pubs.html`). He has received many highly recognized best paper awards on this topic, e.g., the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems Donald O. Pederson Best Paper Award in 2015. He is a Fellow of IEEE and a Fellow of ACM.

Prof. Tsung-Yi Ho is a leading scientist in microfluidic integration and has published more than 150 papers on this topic (`http://www.cs.nthu.edu.tw/~tyho/publications.html`). He has been the recipient of the Humboldt Research Fellowship (2012–2013), which he spent in TUM. He has also received the Hans Fischer Fellowship from TUM-IAS.

In the collaboration, I will focus on architectural synthesis of microfluidic biochips from the functional perspective. Prof. Chakrabarty will analyze new architectures from the test view to evaluate their reliability and testability. Prof. Ho will provide support for design rules for the automated design flow. Both professors have visited TUM with funding support from the Alexander-von-Humboldt-Stiftung and from the Institute for Advanced Study at TUM. They will continue to visit us in the coming years to strengthen our collaboration.

During the period of this project, knowledge exchange with researchers in the microfluidics community, e.g., Prof. Dr. Oliver Hayden in the Department of Electrical and Computer Engineering, TUM, who is working on microfluidics extensively covering characterization, diagnostic, and treatment platform technologies in biomedical science, will be conducted often to obtain information of microfluidic applications and feedback on the proposed framework.

3 Bibliography

[1] J. M. Perkel, ‘‘Microfluidics: Bringing new things to life science,’’ *Science*, vol. 322, no. 5903, pp. 975–977, 2008.

[2] E. Verpoorte and N. F. D. Rooij, ‘‘Microfluidics meets MEMS,’’ *Proc. IEEE*, vol. 91, no. 6, pp. 930–953, 2003.

[3] J. Melin and S. Quake, ‘‘Microfluidic large-scale integration: the evolution of design rules for biological automation,’’ *Annu. Rev. Biophys. Biomol. Struct.*, vol. 36, pp. 213–231, 2007.

[4] K. S. Elvira, X. C. i Solvas, R. C. R. Wootton, and A. J. deMello, ‘‘The past, present and potential for microfluidic reactor technology in chemical synthesis,’’ *Nature Chemistry*, no. 5, pp. 905–915, 2013.

[5] V. Studer, G. Hang, A. Pandolfi, M. Ortiz, W. F. Anderson, and S. R. Quake, ‘‘Scaling properties of a low-actuation pressure microfluidic valve,’’ *J. Appl. Phys.*, vol. 95, no. 1, pp. 393–398, 2004.

[6] [Online]. Available: http://tinyurl.com/biochip-mixing-store

[7] Qiagen, Inc., ‘‘QIAGEN RNase Inhibitor,’’ 2014. [Online]. Available: http://www.qiagen.com

[8] J. M. Perkel, ‘‘Microfluidics: Bringing new things to life science,’’ *Science*, vol. 322, no. 5903, pp. 975–977, 2008.

[9] I. E. Araci and S. R. Quake, ‘‘Microfluidic very large scale integration (mVLSI) with integrated micromechanical valves,’’ *Lab Chip*, vol. 12, pp. 2803–2806, 2012.

[10] Illumina. [Online]. Available: http://www.illumina.com/

[11] International Technology Roadmap for Semiconductors. [Online]. Available: http://www.itrs2.net/

[12] G. D. Micheli, *Synthesis and Optimization of Digital Circuits*. McGraw-Hill Higher Education, 1994.

[13] K. Chakrabarty, R. B. Fair, and J. Zeng, ‘‘Design tools for digital microfluidic biochips: Toward functional diversification and more than moore,’’ *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. (TCAD)*, vol. 29, no. 7, pp. 1001–1017, 2010.

[14] P. Pop, I. E. Araci, and K. Chakrabarty, ‘‘Continuous-flow biochips: Technology, physical-design methods, and testing,’’ *IEEE Design & Test*, vol. 32, no. 6, pp. 8–19, 2015.

[15] M. A. Unger, H.-P. Chou, T. Thorsen, A. Scherer, and S. R. Quake, ‘‘Monolithic microfabricated valves and pumps by multilayer soft lithography,’’ *Science*, vol. 288, no. 5463, pp. 113–116, 2000.

[16] R. Mathies, W. Grover, and E. Jensen, ‘‘Multiplexed latching valves for microfluidic devices and processors,’’ Aug. 2010, US Patent 7,766,033.

[17] N. Amin, W. Thies, and S. P. Amarasinghe, ‘‘Computer-aided design for microfluidic chips based on multilayer soft lithography,’’ in *Proc. IEEE Int. Conf. Comput. Des. (ICCD)*, 2009, pp. 2–9.

[18] W. H. Minhass, P. Pop, J. Madsen, and F. S. Blaga, ‘‘Architectural synthesis of flow-based microfluidic large-scale integration biochips,’’ in *Proc. Int. Conf. on Compilers, Architecture, and Synthesis for Embed. Sys.*, 2012, pp. 181–190.

[19] C. Lin, C. Liu, I. Chen, D. T. Lee, and T. Ho, ‘‘An efficient bi-criteria flow channel routing algorithm for flow-based microfluidic biochips,’’ in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2014, pp. 141:1–141:6.

[20] K. Hu, T. Ho, and K. Chakrabarty, ‘‘Wash optimization and analysis for cross-contamination removal under physical constraints in flow-based microfluidic biochips,’’ *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 35, no. 4, pp. 559–572, 2016.

[21] W. H. Minhass, P. Pop, J. Madsen, and T. Ho, ‘‘Control synthesis for the flow-based microfluidic large-scale integration biochips,’’ in *Proc. IEEE Asia and South Pacific Des. Autom. Conf. (ASP-DAC)*, 2013, pp. 205–212.

[22] K. Hu, T. A. Dinh, T. Ho, and K. Chakrabarty, ‘‘Control-layer routing and control-pin minimization for flow-based microfluidic biochips,’’ *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. (TCAD)*, vol. 36, no. 1, pp. 55–68, 2017.

[23] H. Yao, Q. Wang, Y. Ru, Y. Cai, and T. Ho, ‘‘Integrated flow-control codesign methodology for flow-based microfluidic biochips,’’ *IEEE Design & Test*, vol. 32, no. 6, pp. 60–68, 2015.

[24] H. Yao, T. Ho, and Y. Cai, ‘‘PACOR: practical control-layer routing flow with length-matching constraint for flow-based microfluidic biochips,’’ in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2015, pp. 142:1–142:6.

[25] A. M. Amin, M. Thottethodi, T. N. Vijaykumar, S. Wereley, and S. C. Jacobson, ‘‘Automatic volume management for programmable microfluidics,’’ in *Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation*, 2008, pp. 56–67.

[26] D. Mitra, S. Roy, S. Bhattacharjee, K. Chakrabarty, and B. B. Bhattacharya, ‘‘On-chip sample preparation for multiple targets using digital microfluidics,’’ *IEEE J. Technol. Comput. Aided Design*, vol. 33, no. 8, pp. 1131–1144, 2014.

[27] T.-M. Tseng, B. Li, U. Schlichtmann, and T.-Y. Ho, ‘‘Storage and caching: Synthesis of flow-based microfluidic biochips,’’ *IEEE Design & Test*, vol. 32, no. 6, pp. 69–75, 2015.

[28] T.-M. Tseng, B. Li, M. Li, T.-Y. Ho, and U. Schlichtmann, ‘‘Reliability-aware synthesis with dynamic device mapping and fluid routing for flow-based microfluidic biochips,’’ *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. (TCAD)*, vol. 35, no. 12, pp. 1981–1994, 2016.

[29] C. Liu, B. Li, B. B. Bhattacharya, K. Chakrabarty, T.-Y. Ho, and U. Schlichtmann, ‘‘Testing microfluidic fully programmable valve arrays (FPVAs),’’ in *Proc. IEEE Design, Autom., and Test Europe Conf. (DATE)*, 2017, pp. 91–96.

[30] C. Liu, B. Li, T.-Y. Ho, K. Chakrabarty, and U. Schlichtmann, ‘‘Design-for-testability for continuous-flow microfluidic biochips,’’ in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2018, pp. 164:1–164:6.

[31] A. Bernardini, C. Liu, B. Li, and U. Schlichtmann, ‘‘Fault localization in programmable microfluidic devices,’’ in *Proc. IEEE Design, Autom., and Test Europe Conf. (DATE)*, 2019.

[32] Y. Zhu, B. Li, T.-Y. Ho, Q. Wang, H. Yao, R. Wille, and U. Schlichtmann, ‘‘Multi-channel and fault-tolerant control multiplexing for flow-based microfluidic biochips,’’ in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, 2018, pp. 123:1–123:8.

[33] P. Spindler, U. Schlichtmann, and F. M. Johannes, ‘‘Kraftwerk2 – a fast force-directed quadratic placement approach using an accurate net model,’’ *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. (TCAD)*, vol. 27, no. 8, pp. 1398–1411, 2008.

[34] G. L. Zhang, B. Li, and U. Schlichtmann, ‘‘EffiTest: Efficient delay test and statistical prediction for configuring post-silicon tunable buffers,’’ in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2016, pp. 60:1–60:6.

[35] M. Barke and U. Schlichtmann, ‘‘A cross-layer approach to measure the robustness of integrated circuits,’’ *ACM J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 3, pp. 24:1–24:22, 2015.

[36] B. Li, N. Chen, Y. Xu, and U. Schlichtmann, ‘‘On timing model extraction and hierarchical statistical timing analysis,’’ *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. (TCAD)*, vol. 32, no. 3, pp. 367–380, 2013.

[37] M. Ibrahim, K. Chakrabarty, and U. Schlichtmann, ‘‘Synthesis of a cyberphysical hybrid microfluidic platform for single-cell analysis,’’ *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. (TCAD)*, 2019.

4 Requested modules/funds

4.1 Basic Module

4.1.1 Funding for Staff

- One full-time research assistant (PhD candidate or postdoctoral researcher), TV-L E13, for three years

Since this is an interdisciplinary research topic, an extensive understanding of design automation for integrated circuits and biochemical systems is required. In addition, the project provides a good chance for the EDA research community to reach out to other research fields. Accordingly, it is applied that this position can be filled by a postdoctoral researcher.

- Two work student positions (each 40 hours/month) for three years

This interdisciplinary research project plans to develop a design automation framework for microfluidic biochips. It requires that many algorithms from other research fields, e.g., design automation for integrated circuits, are implemented and tested, while considering the unique characteristics of biochips. Therefore, the support from work students is highly expected. Reciprocally, this project provides students a chance to expose themselves to various problems ranging from the basic mechanism of biochips to high-level programming techniques.

4.1.2 Direct Project Costs

4.1.2.1 Equipment up to € 10,000, Software and Consumables

None

4.1.2.2 Travel Expenses

Trips for presenting research results at national and international conferences are planned.

Three trips to overseas conferences, e.g., ACM/IEEE Design Automation Conference, USA, IEEE/ACM Conference on Computer-Aided Design, USA: €3,000.00×3

Three trips to European conferences and workshops: €1,500.00×3

Total sum of travel expenses: €13,500.00

4.1.2.3 Visiting Researchers

None

4.1.2.4 Expenses for Laboratory Animals

None

4.1.2.5 Other Costs

None

4.1.2.6 Project-related publication expenses

Journal paper publication with IEEE open access option: €1,900.00

4.1.3 Instrumentation

4.1.3.1 Equipment exceeding Euro 10,000

None

4.1.3.2 Major Instrumentation exceeding Euro 50,000

None

4.2 Module Temporary Position for Funding

None

4.3 Module Replacement Funding

None

4.4 Module Temporary Clinician Substitute

None

4.5 Module Mercator Fellows

None

4.6 Module Workshop Funding

None

4.7 Module Public Relations Funding

None

5 Project requirements

5.1 Employment status information

PD Dr.-Ing. habil. Bing Li, Akademischer Rat auf Zeit bis 31.07.2021, Institute for Electronic Design Automation, Technical University of Munich

5.2 First-time proposal data

This is the first-time proposal from PD Dr.-Ing. habil. Bing Li.

5.3 Composition of the project group

Besides the research assistant supported by this project, other members in the research group on emerging systems in the Institute for Electronic Design Automation, TUM, will work on related topics ranging from test of microfluidic biochips to optical interconnect systems. These members include Prof. Ulf Schlichtmann and Ms. Ying Zhu (PhD candidate, supported by TUM IGSSE scholarship). These members will provide necessary technical support to this project, specially the knowledge of existing design methodologies and algorithms.

5.4 Cooperation with other researchers

5.4.1 Researchers with whom you have agreed to cooperate on this project

Prof. Krishnendu Chakrabarty, Duke University, US
Prof. Tsung-Yi Ho, National Tsing Hua University, Taiwan

5.4.2 Researchers with whom you have collaborated scientifically within the past three years

Within several projects in the past, we have collaborated with other researchers extensively. To be named are the strong collaborations with Prof. Krishnendu Chakrabarty (Duke University, US), Prof. David Z. Pan (UT Austin, US), Prof. Jiang Hu (Texas A&M University, US), Prof. Yiyu Shi (University of Notre Dame, US), Prof. Paul Pop (Technical University of Denmark), Prof. Bhargab B. Bhattacharya (Indian Statistical Institute, India), Prof. Masanori Hashimoto (Osaka University, Japan), Prof. Tsung-Yi Ho (National Tsing Hua University, Taiwan), Prof. Hailong Yao (Tsinghua University, China).

5.5 Scientific equipment

The Institute for Electronic Design Automation is equipped with a central file server and a cluster of computing servers containing Dual-/Quad-Core CPUs running Linux. Therefore, all necessary equipment for running the project is already available.

5.6 Project-relevant cooperation with commercial enterprises

None

5.7 Project-relevant participation in commercial enterprises

None

6 Additional information

None